Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет (институт) Информационных технологий

Кафедра Прикладная математика

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Е. Г. Боровцов

(подпись преподавателя) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет

по лабораторной (практической) работе №3

по дисциплине Виртуализация информационной инфраструктуры

(наименование дисциплины)

ЛР 09.03.04.14.000 ОТ

(обозначение документа)

Студент группы ПИ-61 В.Р.Оверченко

(инициалы, фамилия)

Преподаватель Е. Г. Боровцов

(инициалы, фамилия)

Барнаул 2020

Создали виртуальную машину в VirtualBox для Debian 10 со следующими характеристиками:

• Память – 4ГБ;

• 2 процессорных ядра;

• Виртуальный диск – 40 ГБ.

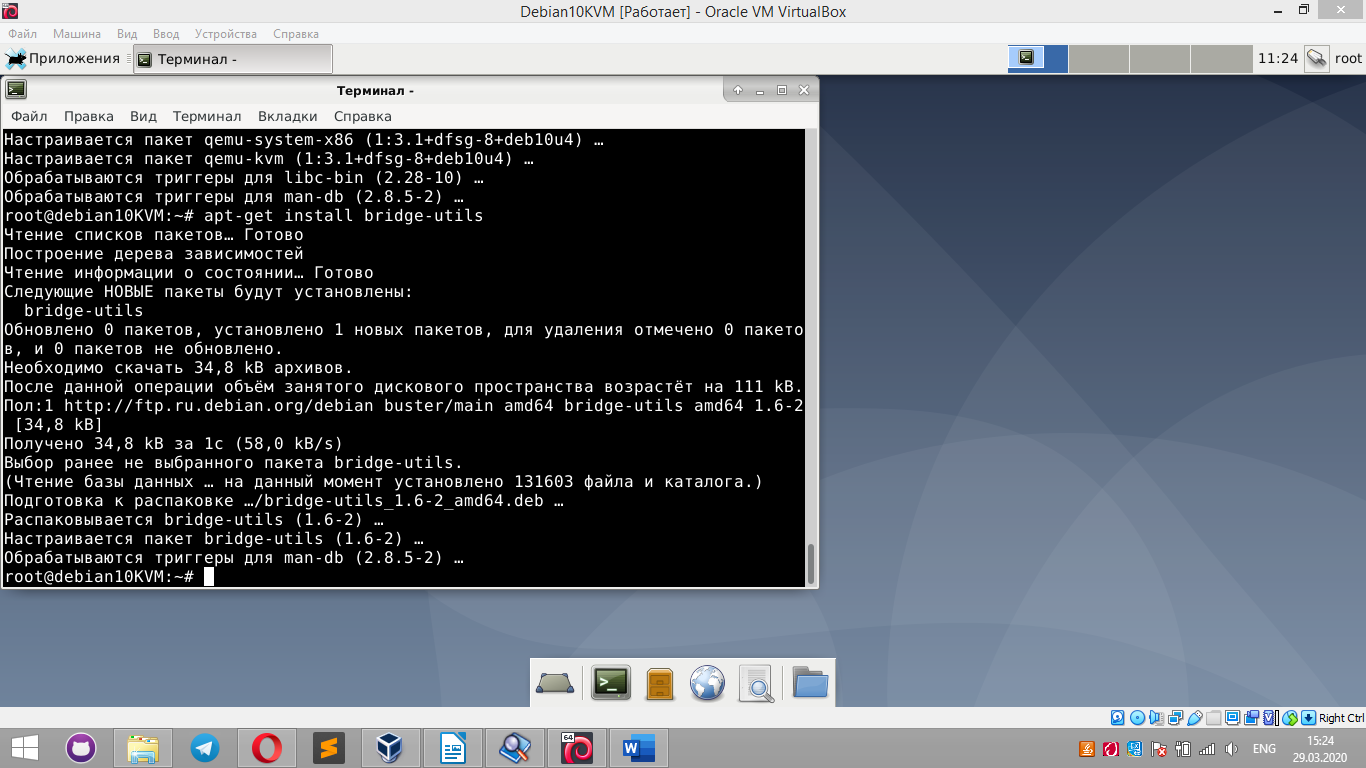
Далее выполнили установку системы Debian 10 как и в прошлой лабораторной работе (скрины добавлять не стали).

После успешной инсталляции системы и дополнений гостевой ОС переходим к установке необходимых для работы пакетов. Кстати, в предыдущей работе дополнения установлены не были. Было выдано сообщение, что не найдены файлы для сборки ядра, которое никто не прочитал. Поэтому перед подключением образа с дополнениями гостевой ОС были установлены пакеты  *build-essential* и *dkms* и только потом подмонтирован диск и произведена установка.

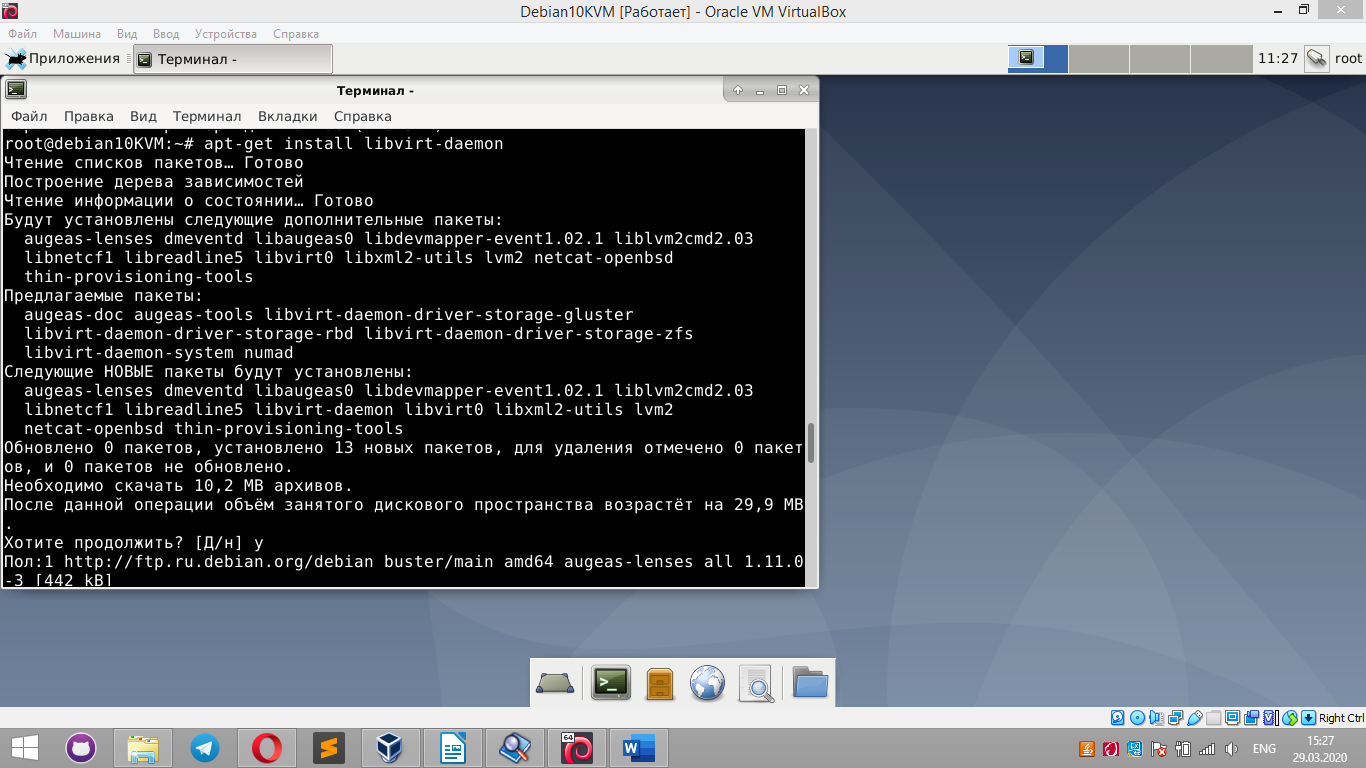
1. Qemu-kvm – гипервизор, поддерживающий аппаратную виртуализацию:



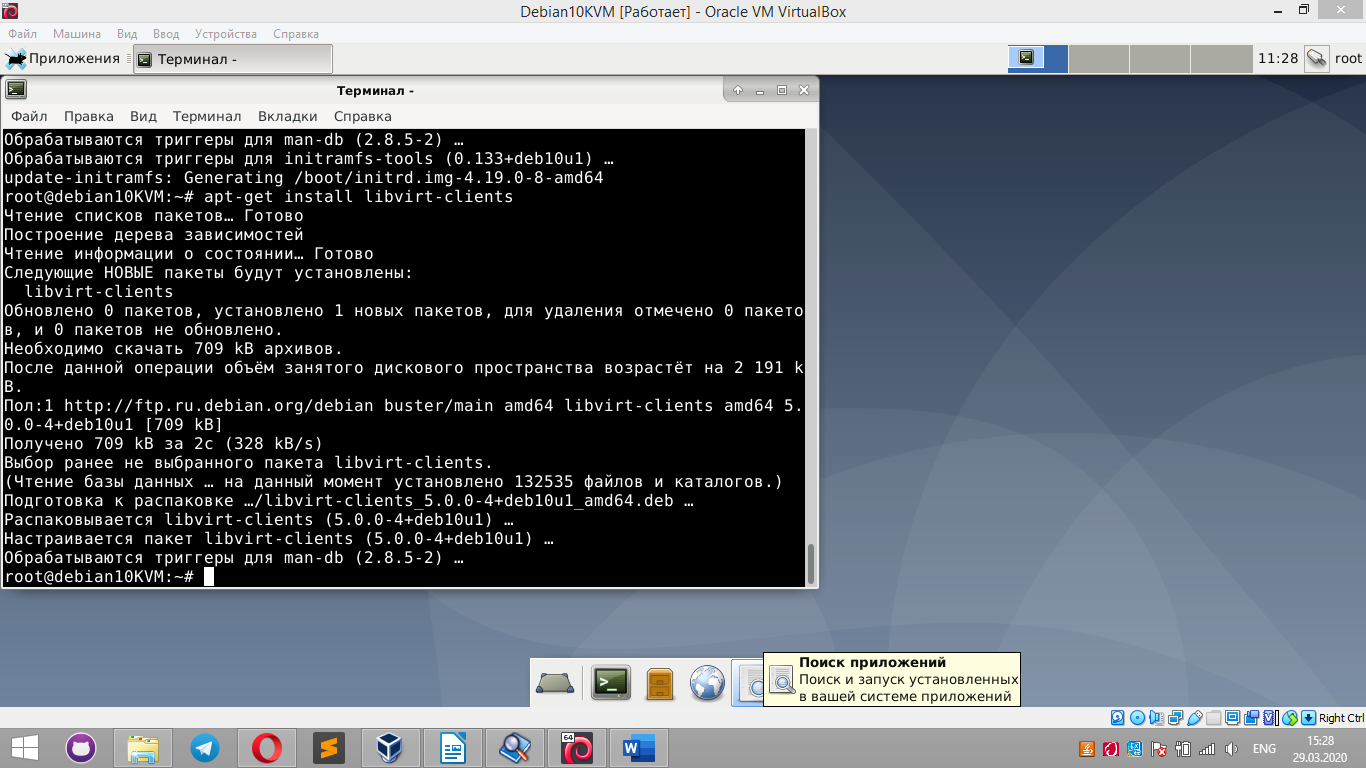
1. Bridge-utils – пакет для настройки сетевого моста для виртуализованных сред:



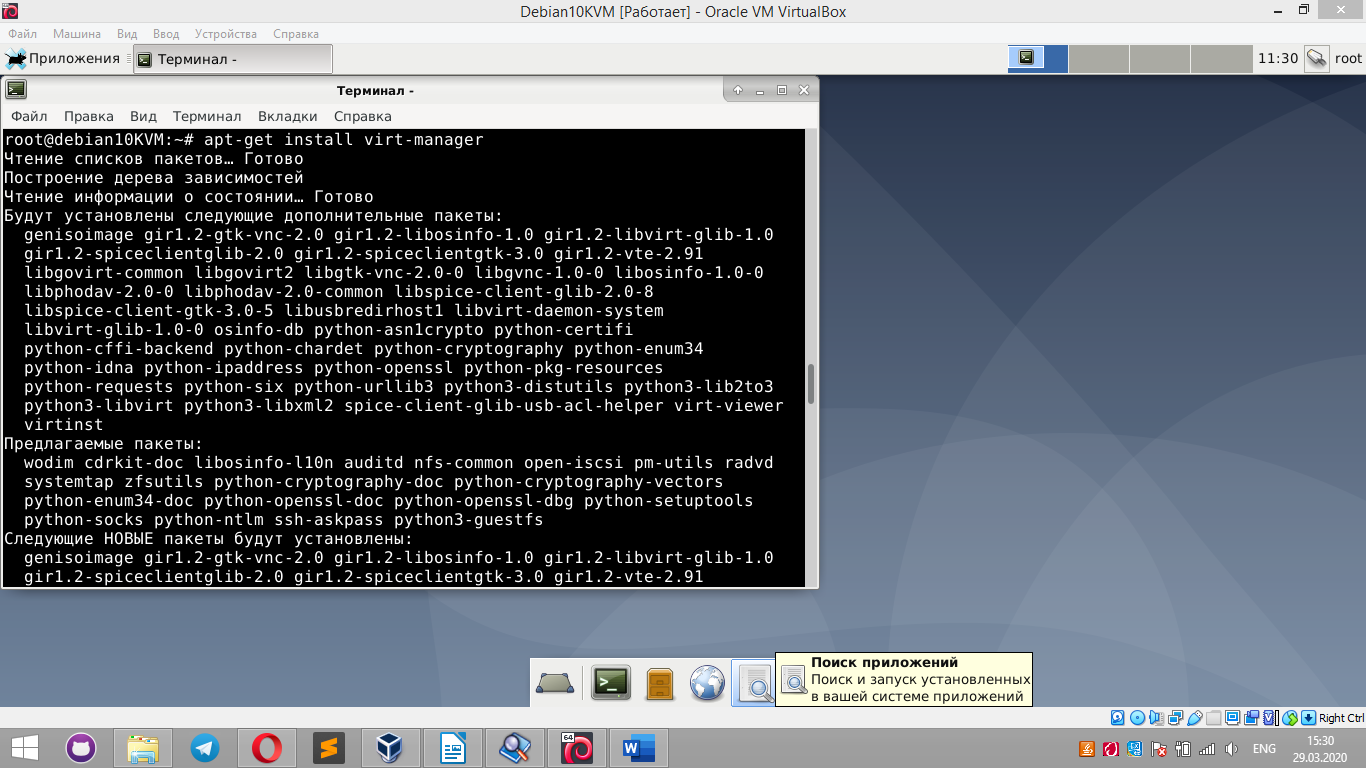
1. Libvirt-daemon - пакет содержит файлы конфигурации для запуска демона libvirt в качестве системной службы:



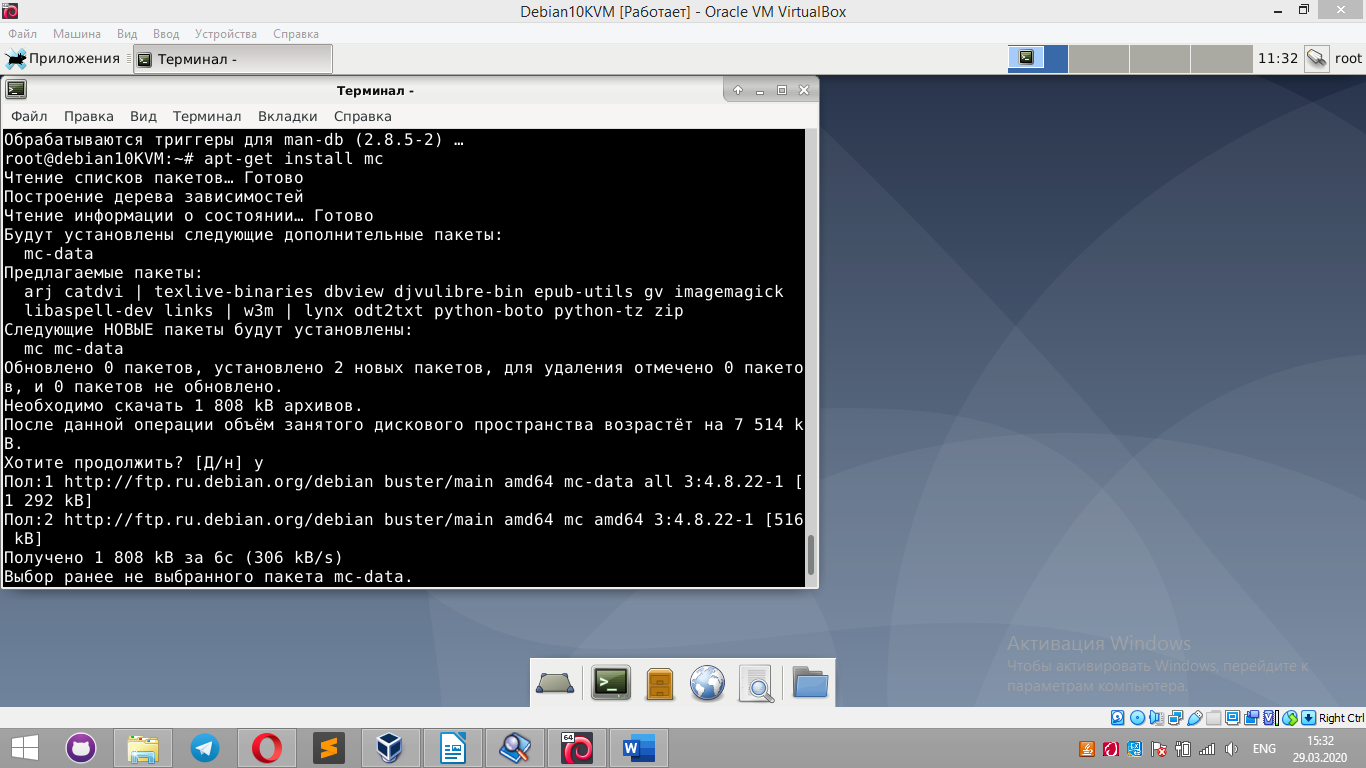
1. Libvirt-clients:



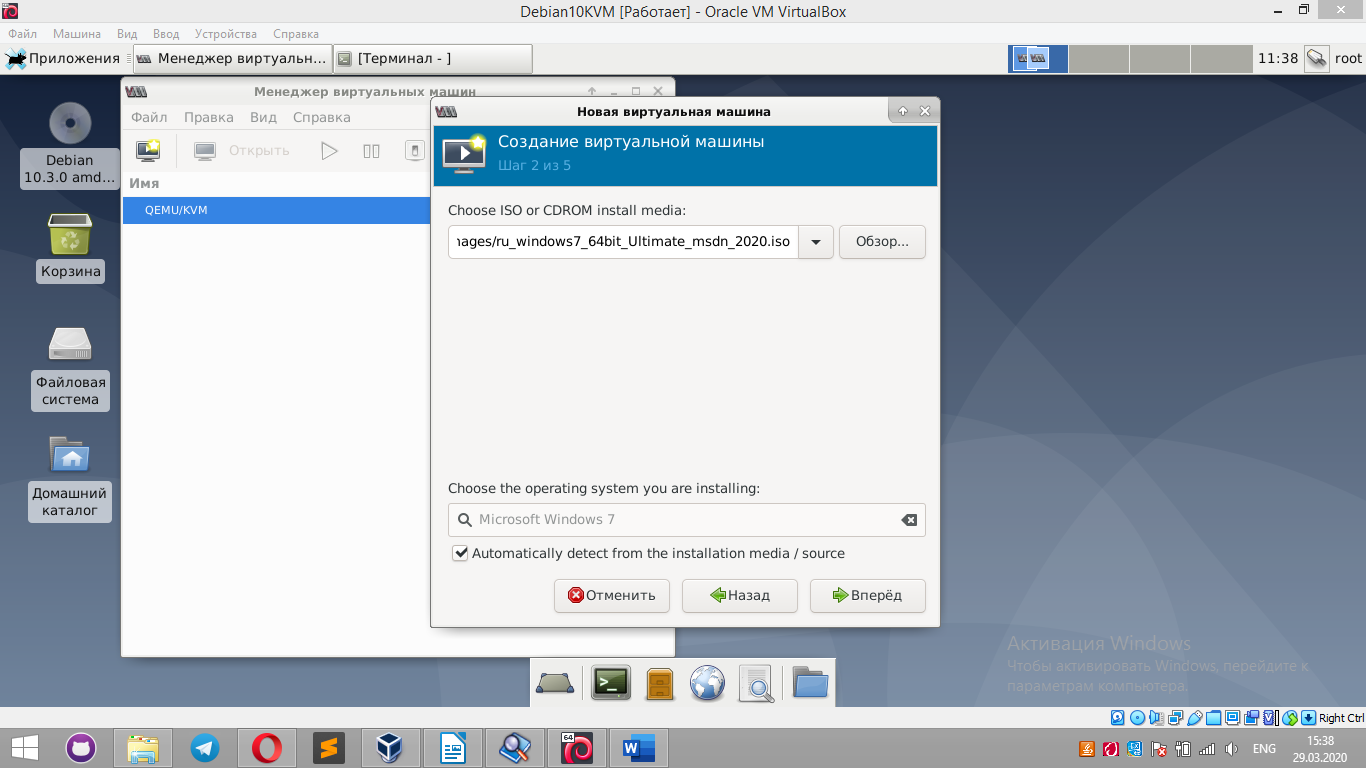
1. Virt-manager - программа для управления и мониторинга KVM:



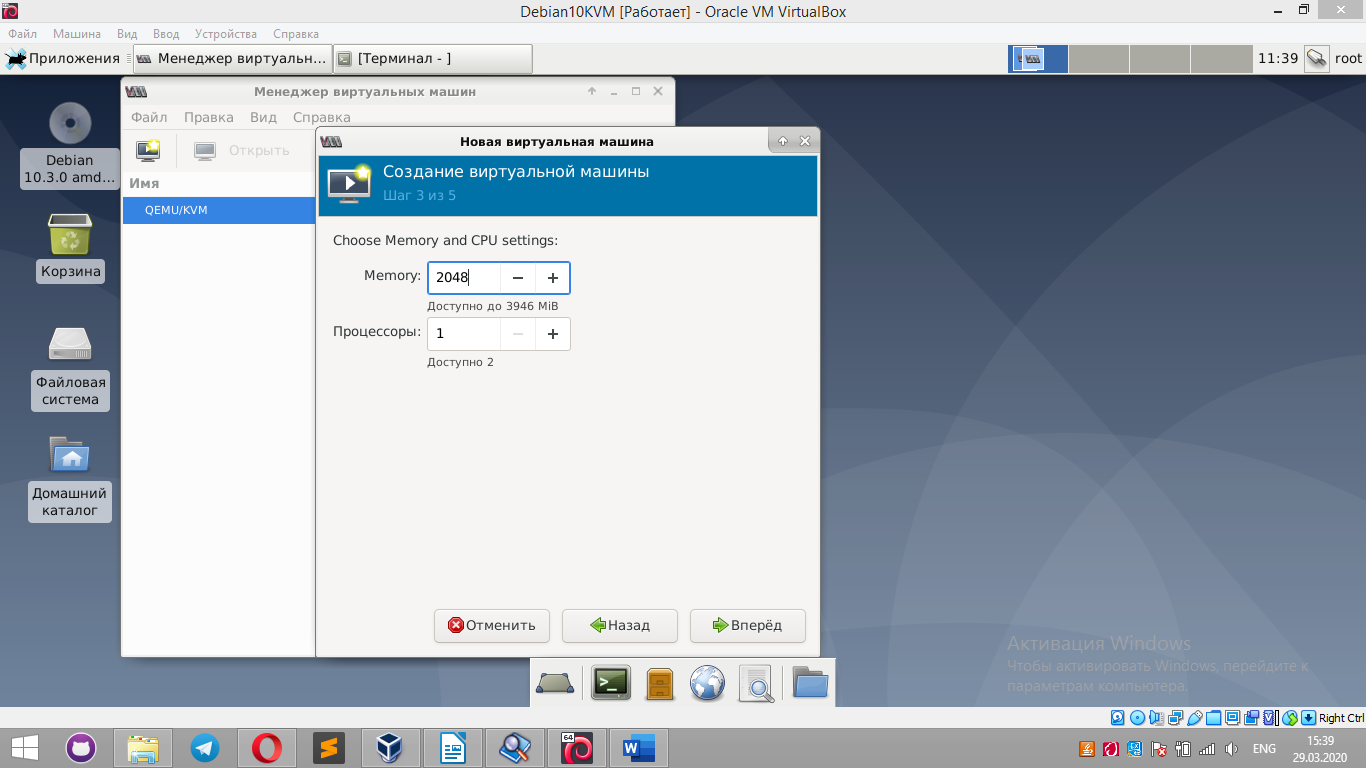
1. Mc – консольный файловый менеджер:



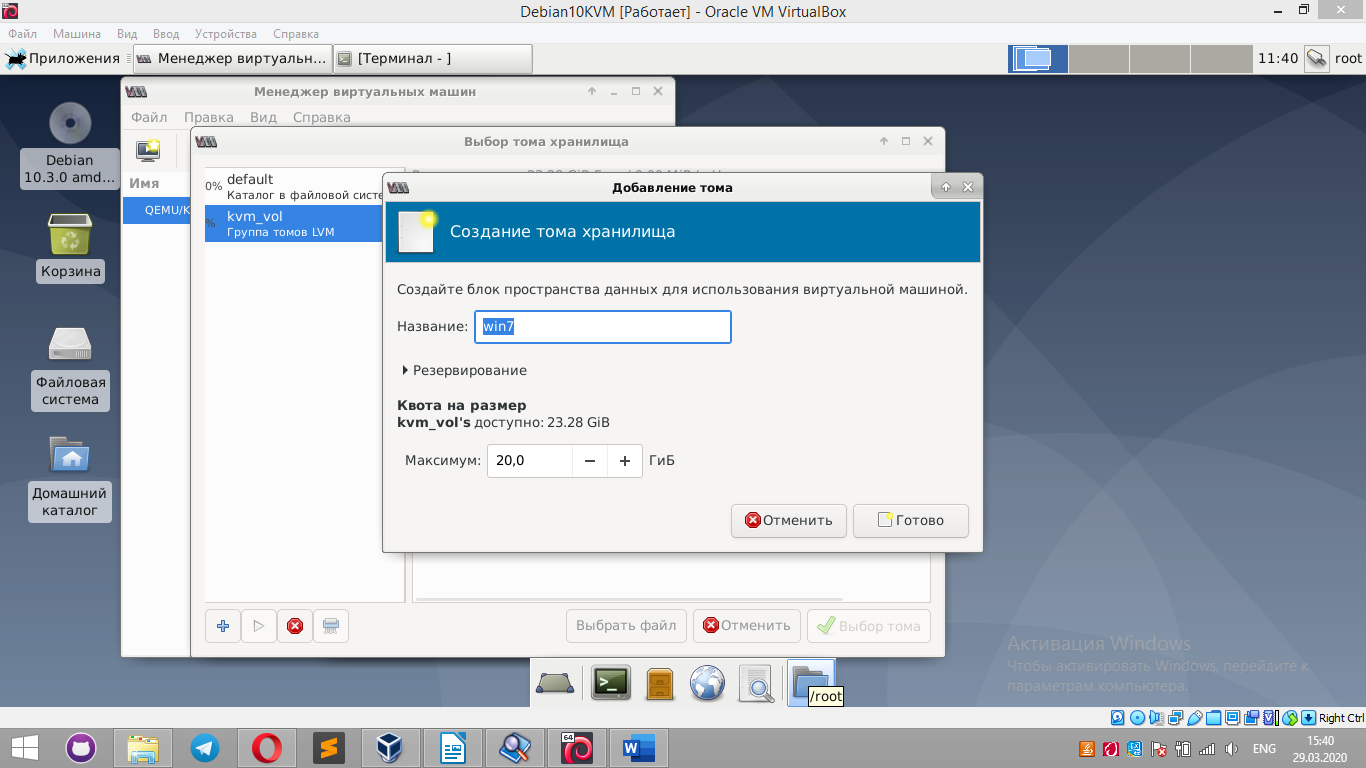
После установки пакетов переходим в менеджер виртуальных машин, где появился нужный нам раздел QEMU/KVM. Создаём новую виртуальную машину с гостевой ОС Windows 7:

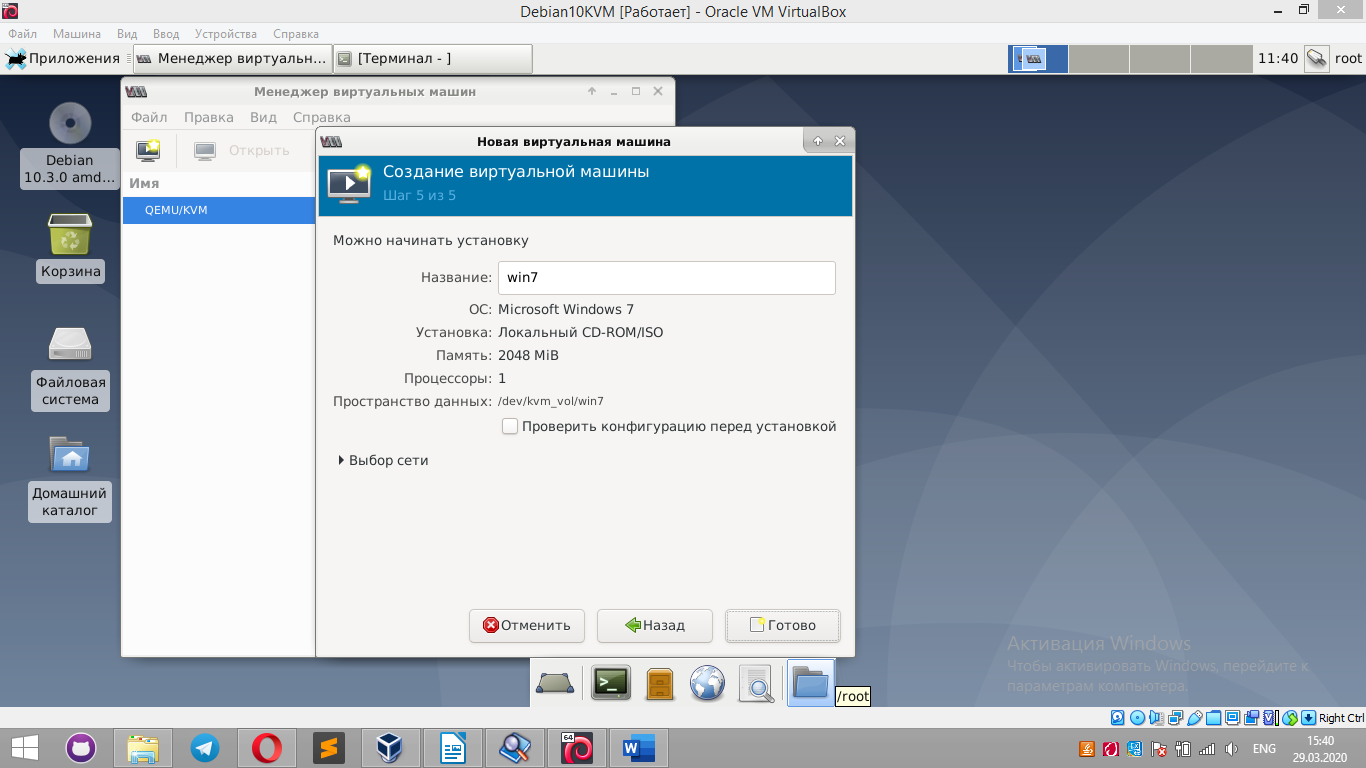


Выделяем для машины 2Gb RAM и 1 процессорное ядро:

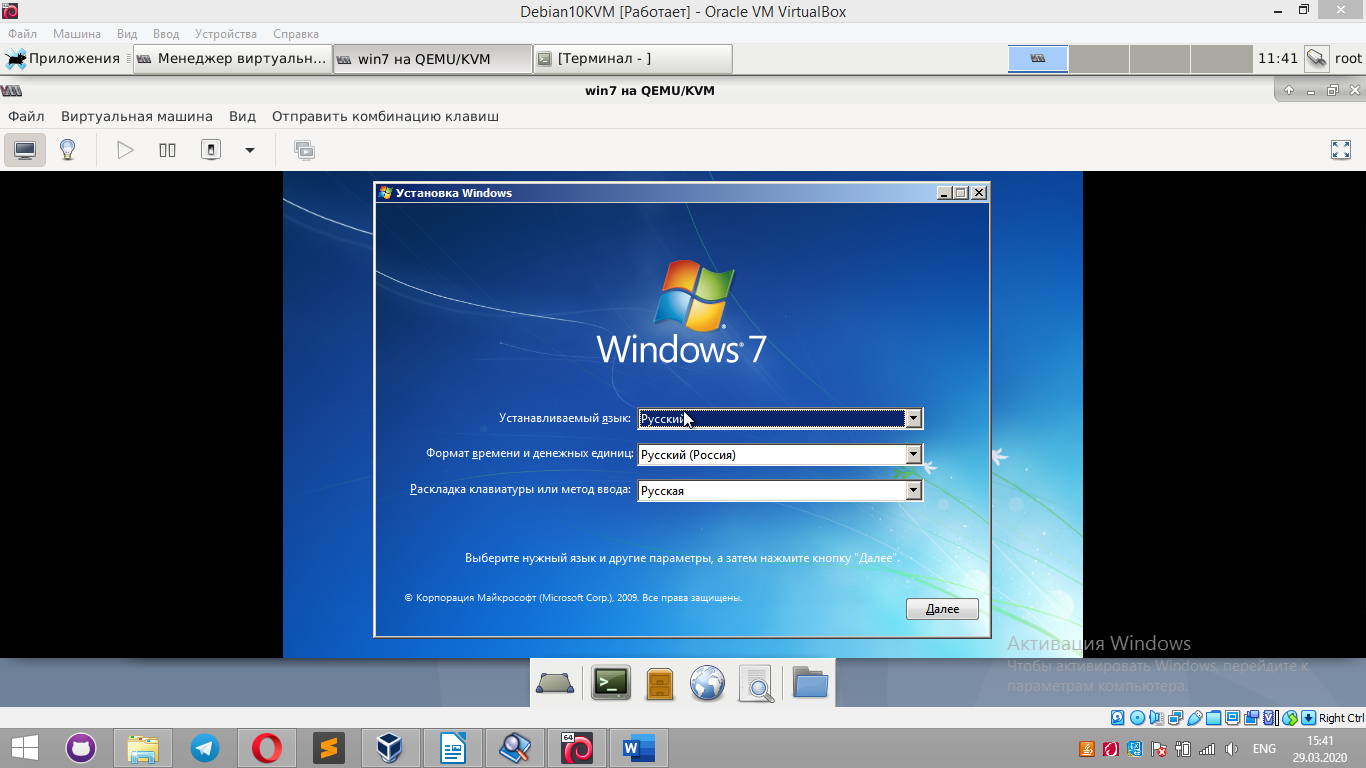


Добавляем том хранилища в заранее созданной группе томов LVM *kvm\_vol.* Группа томов создавалась по тому же принципу, что и в предыдущей лабораторной.

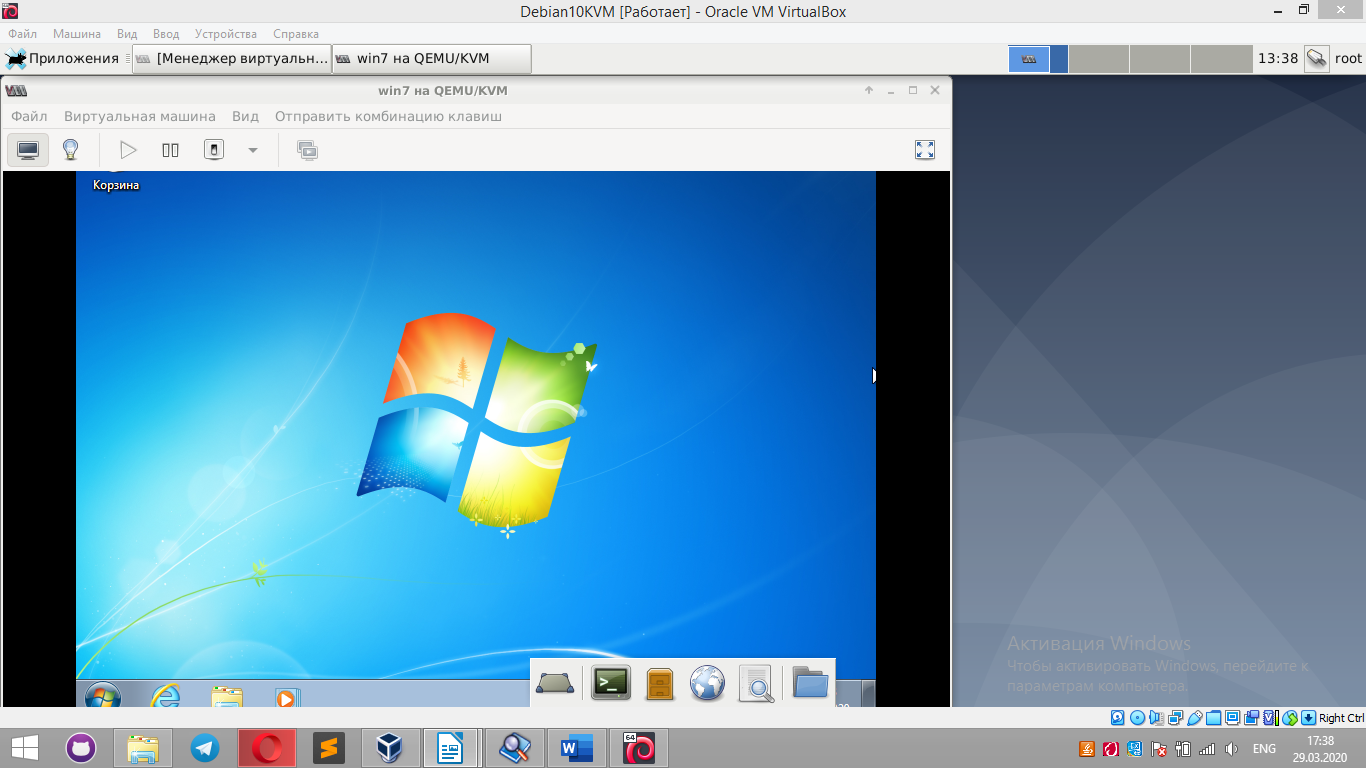




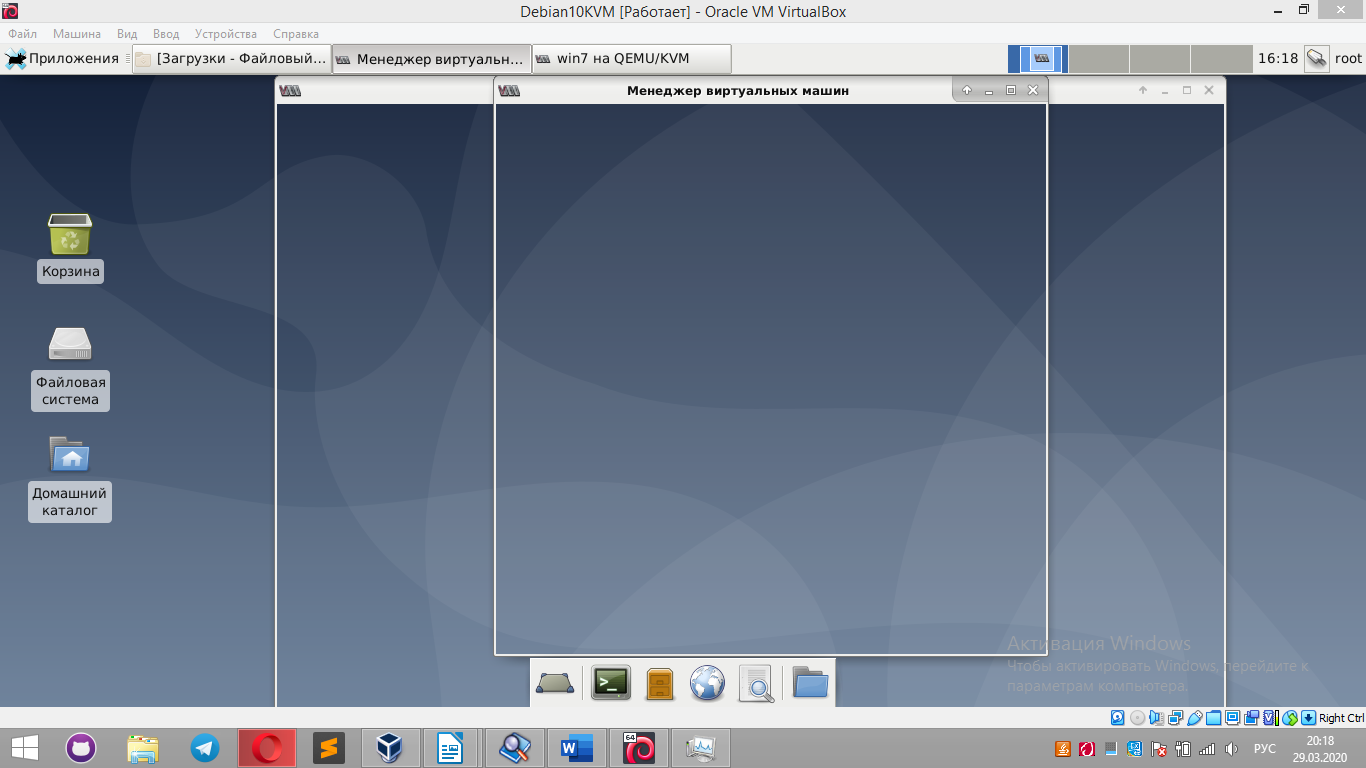
Запускаем инсталляцию винды:



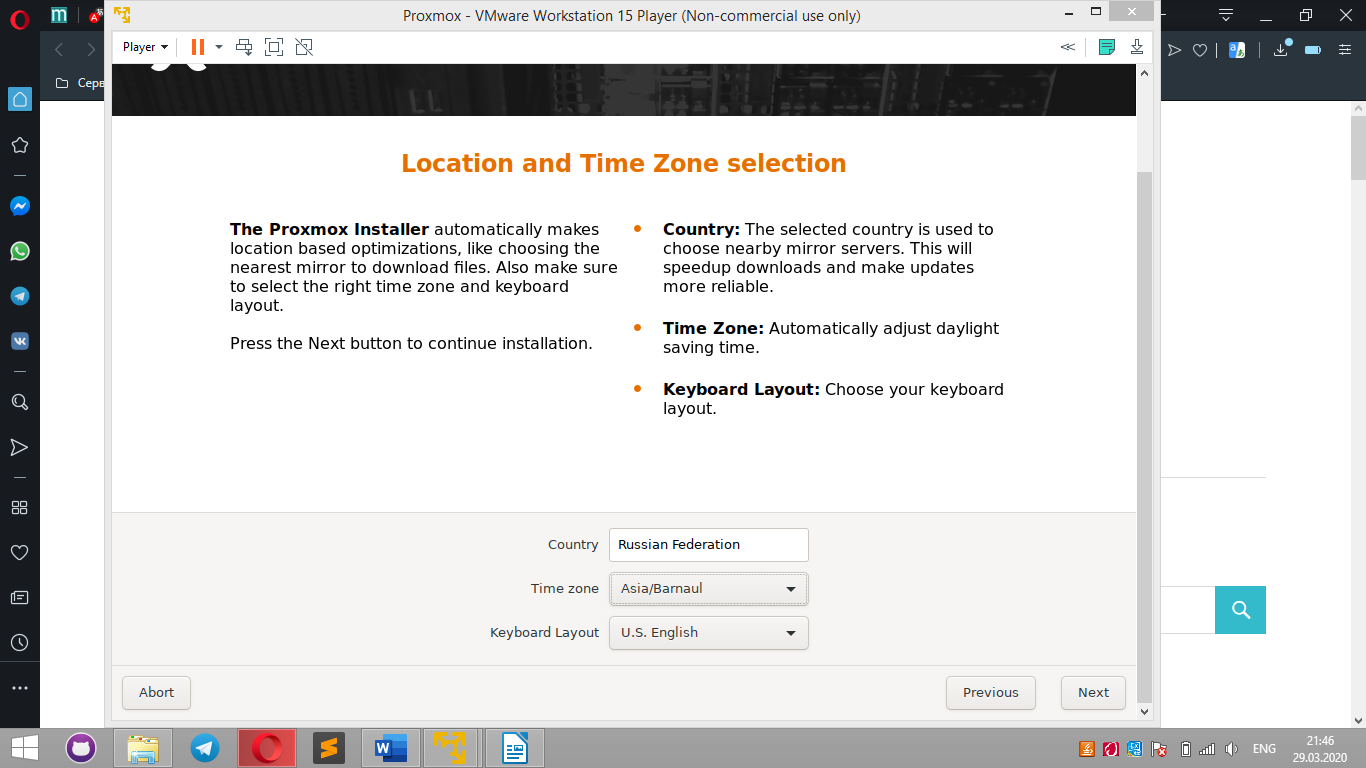
Шаги установки опущены, т.к. не представляют ничего нового. По итогу имеем работающую систему.



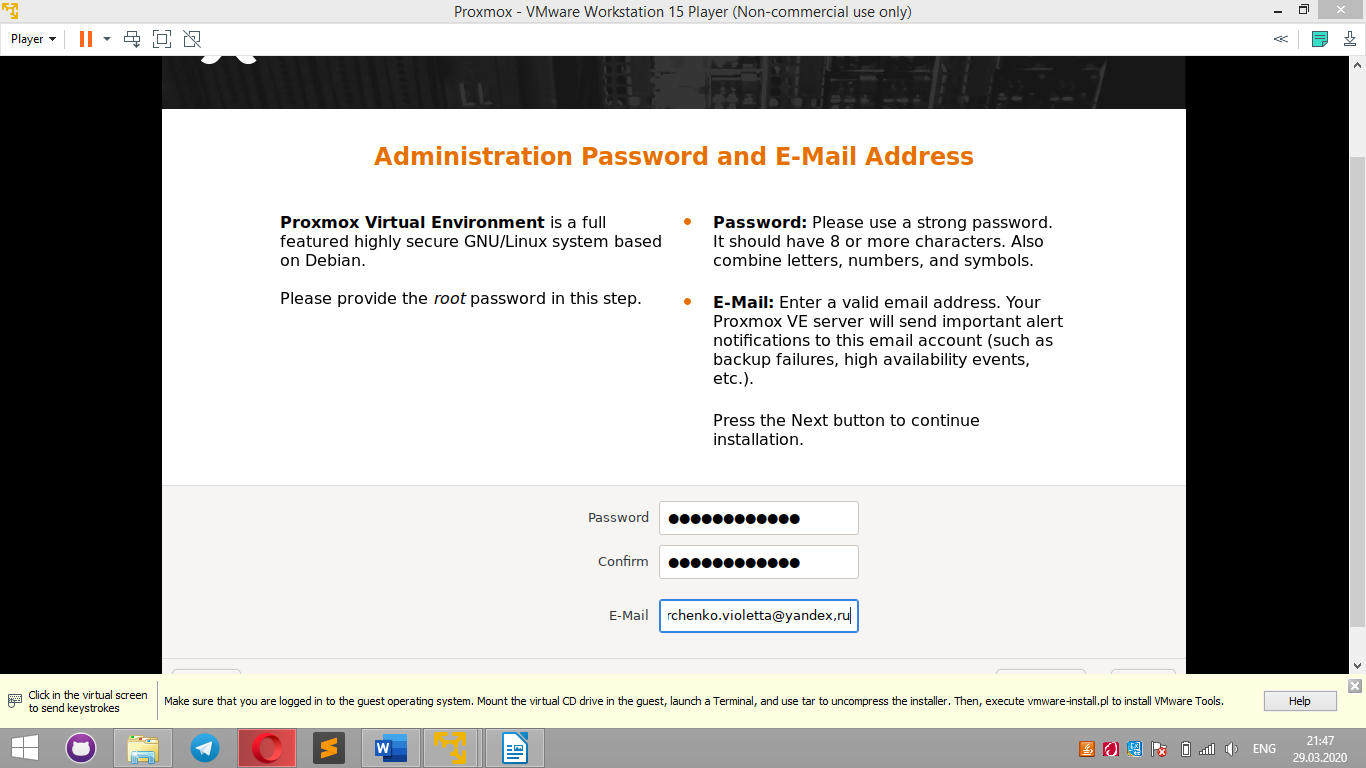
Для нормального функционирования необходимо установить пакет драйверов virtio-win, но этого не удалось сделать. Видимо, из-за высокой нагрузки система переставала откликаться, без перезагрузки ничего нельзя было выполнить. Поэтому было решено оставить всё как есть.



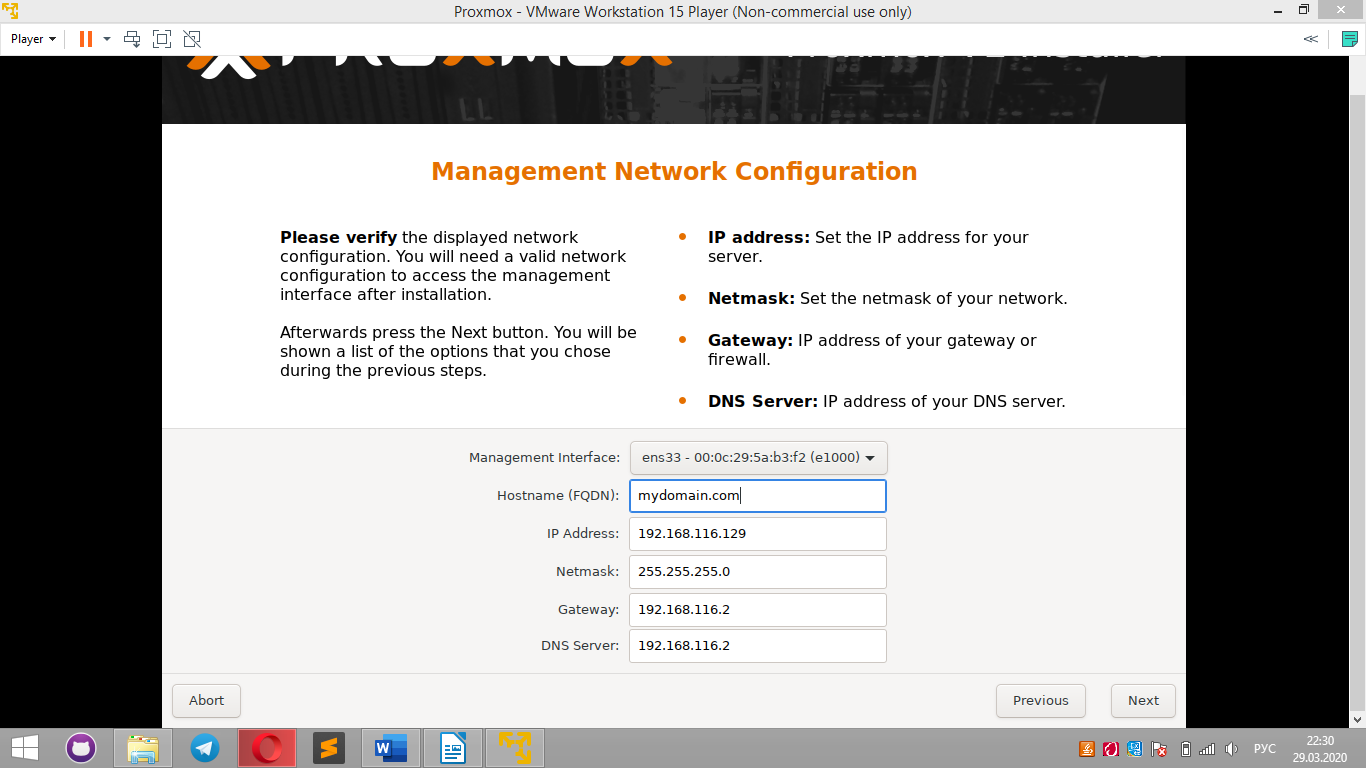
Переходим ко второй части задания. В среде VMware создали виртуальную машину для установки Proxmox со следующими характеристиками: 4GB RAM и виртуальный диск размером 60GB. Запускаем инсталляцию. Выбираем временную зону, раскладку клавиатуры:

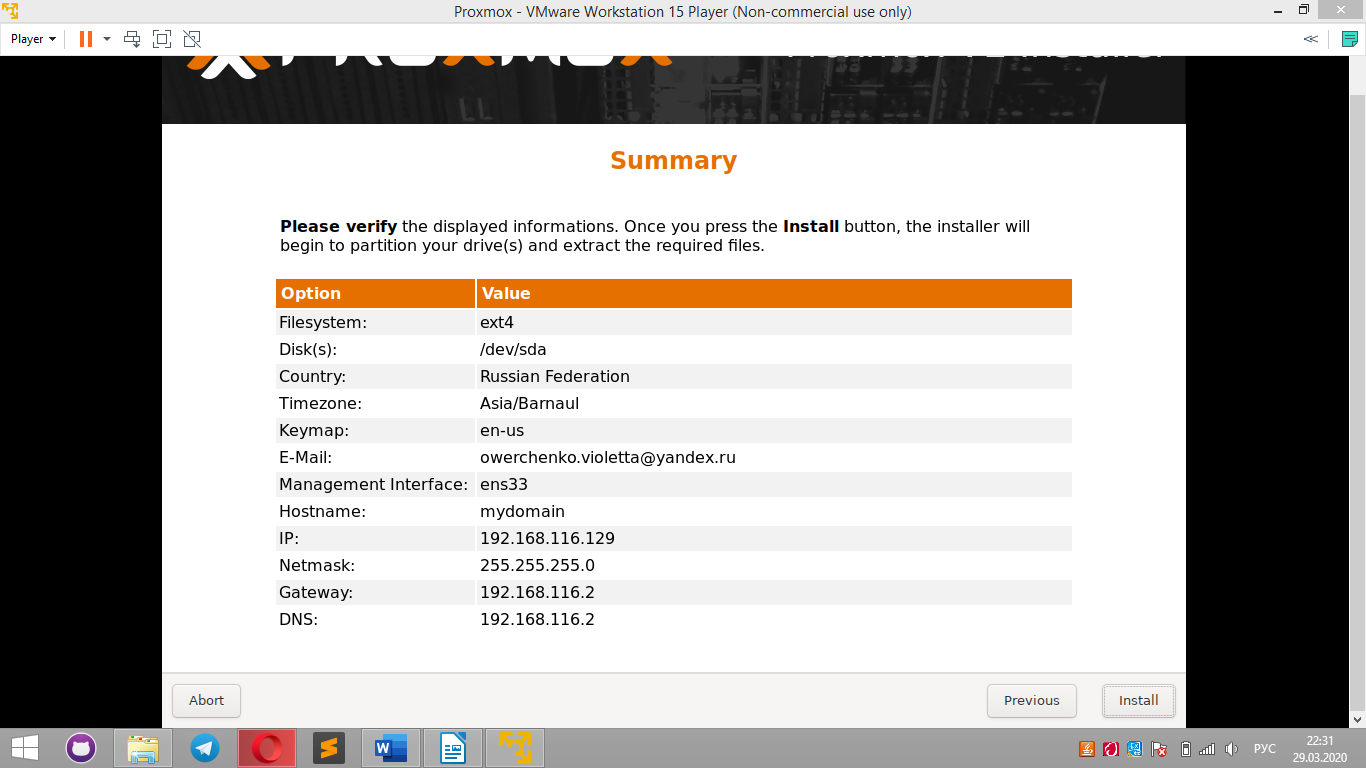


Придумываем пароль для администратора:

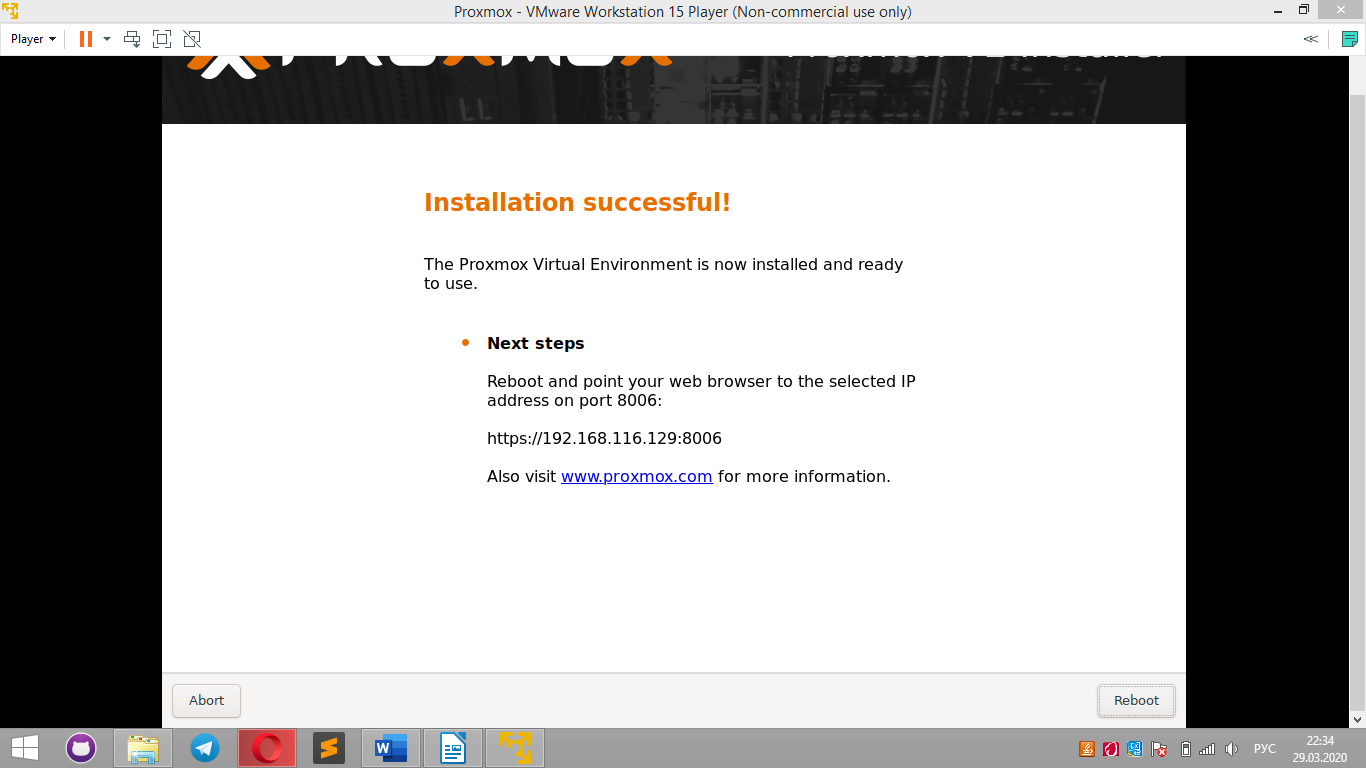


Выполняем конфигурацию сети:

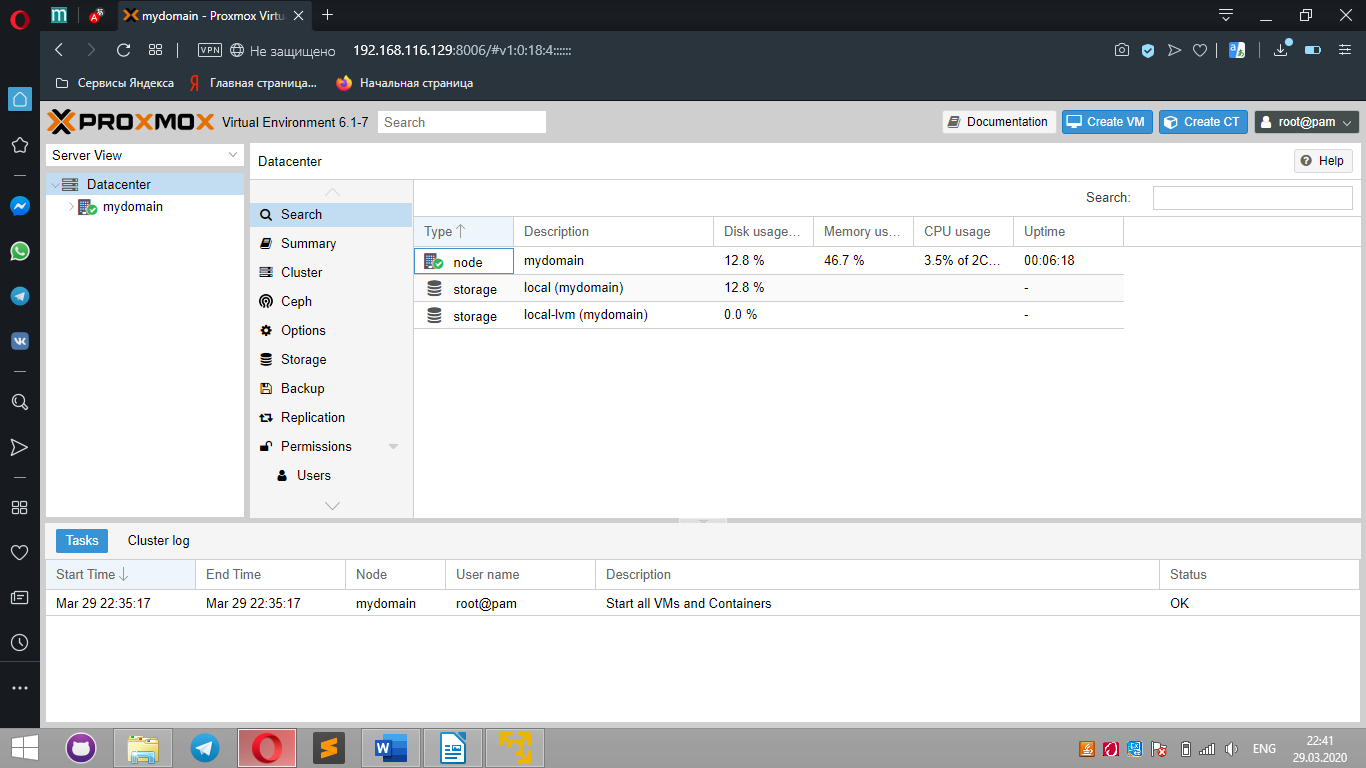




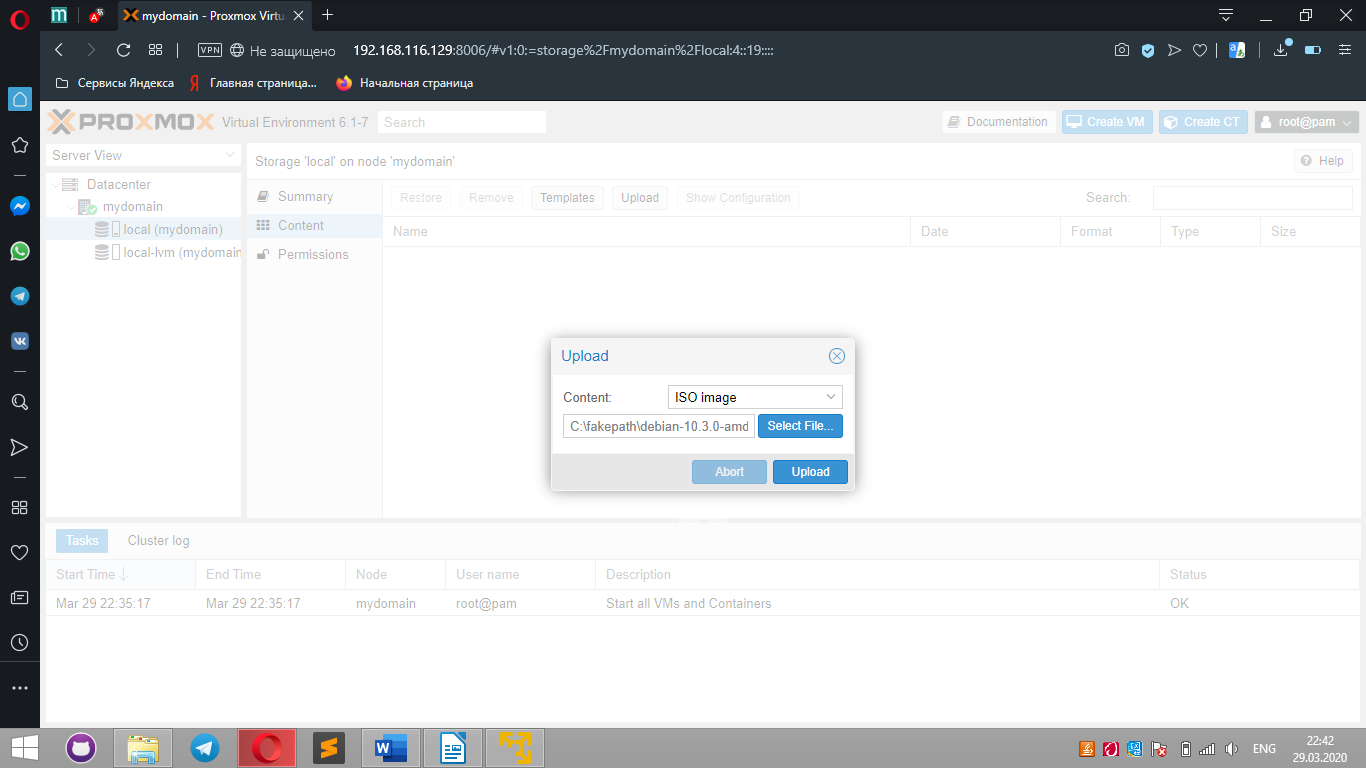
Установка прошла успешно:

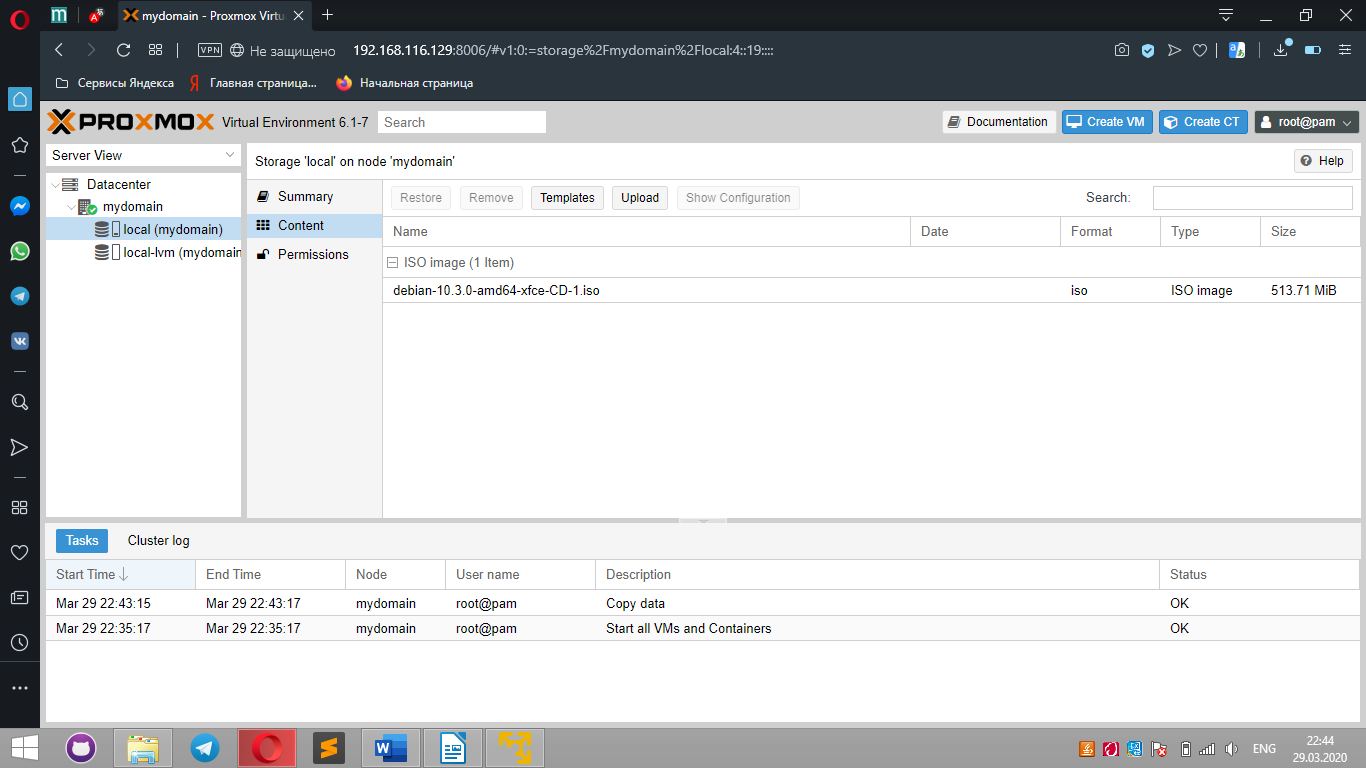


После перезагрузки системы переходим по адресу для управления proxmox через веб-панель. Видим созданный нами домен:

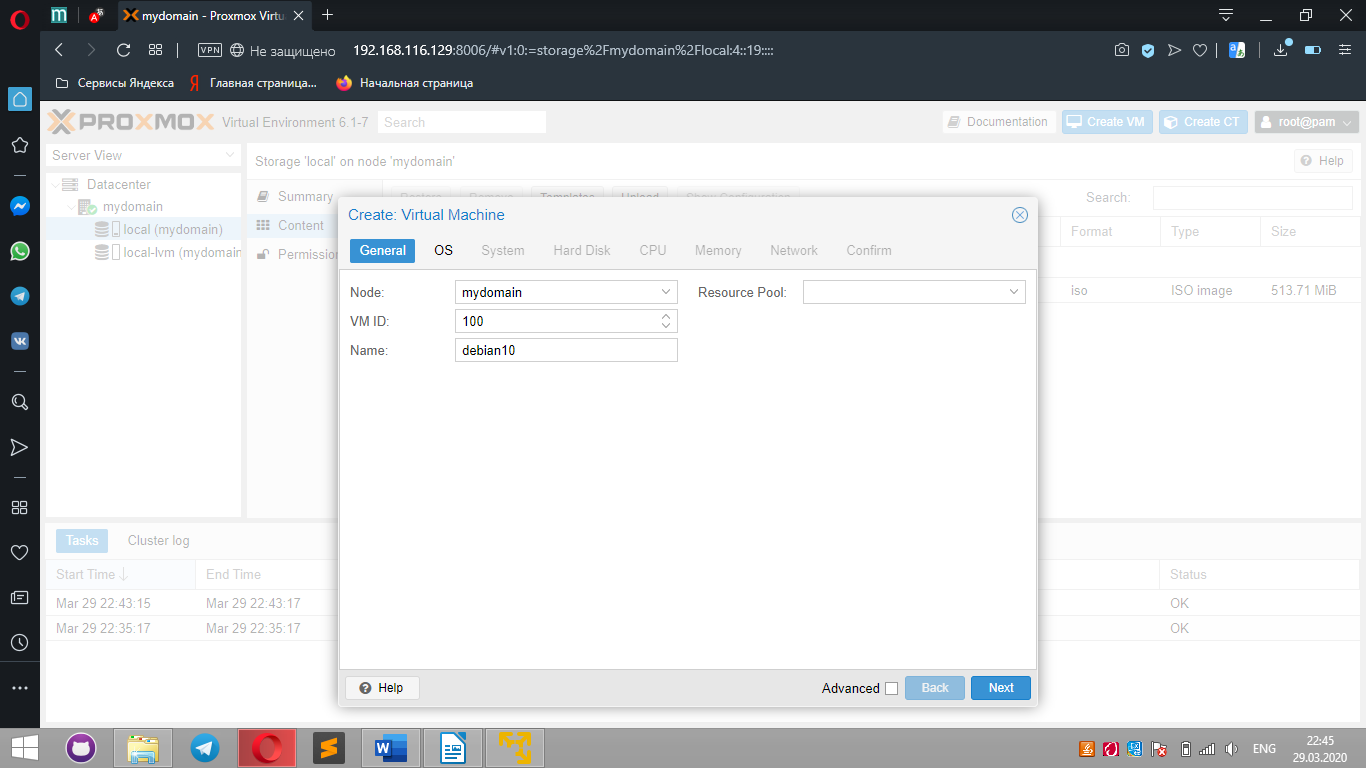


Загружаем iso-образ Debian10 в качестве гостевой ОС:





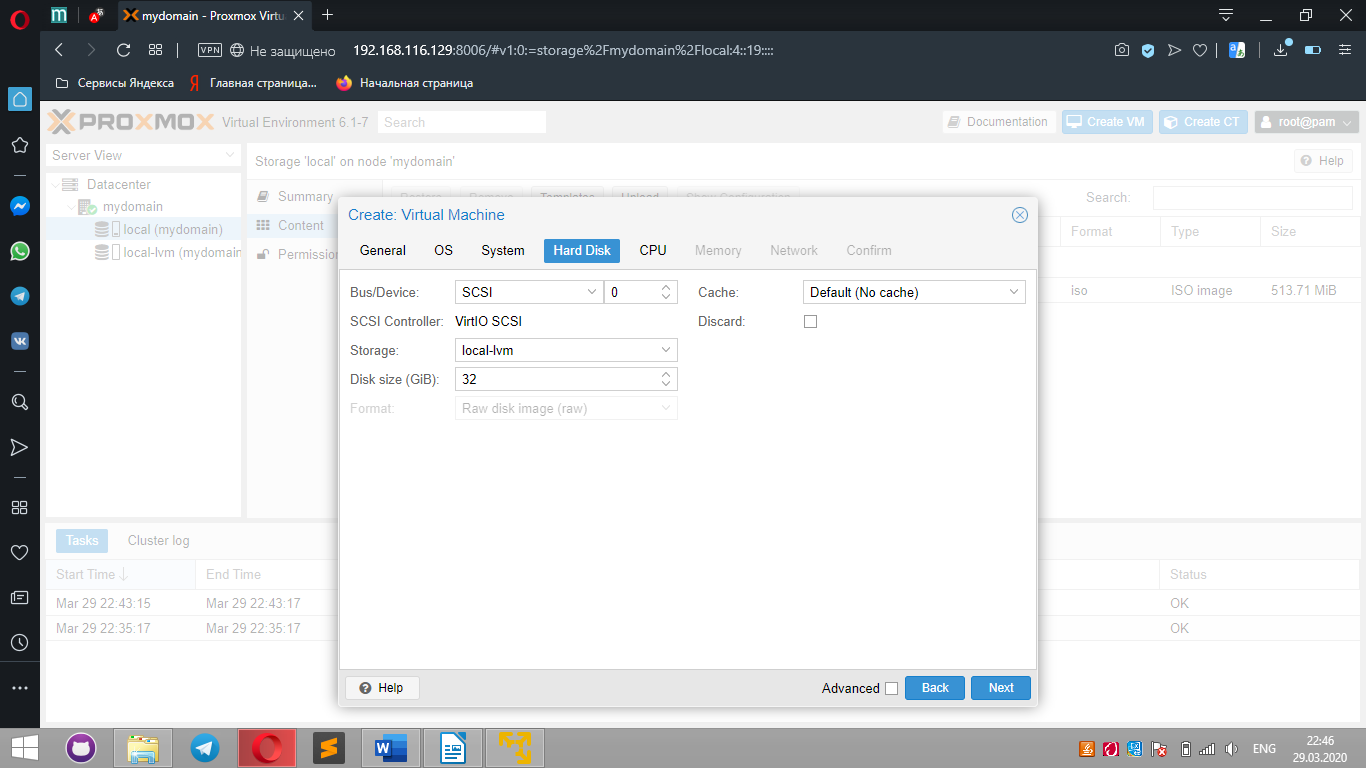
Создаём виртуальную машину:



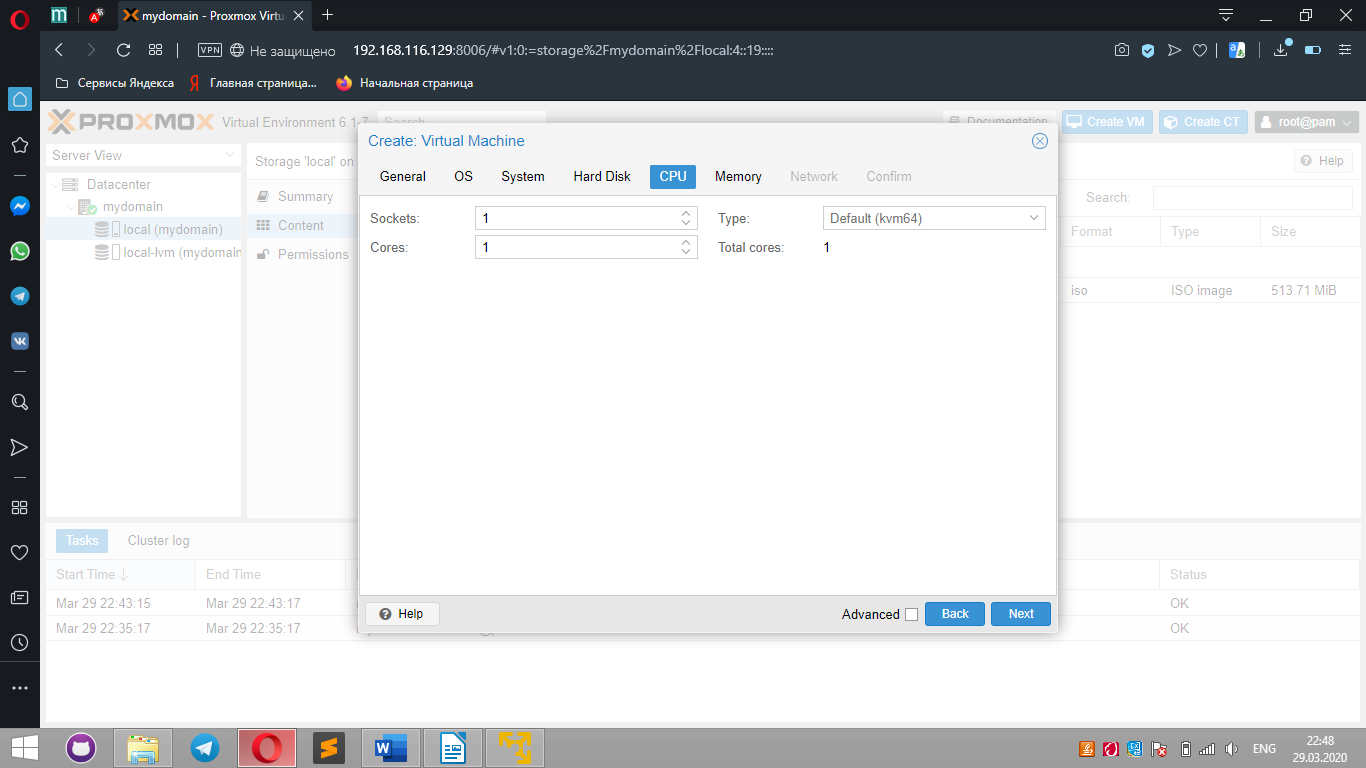
Выбираем загруженный ранее установщик:



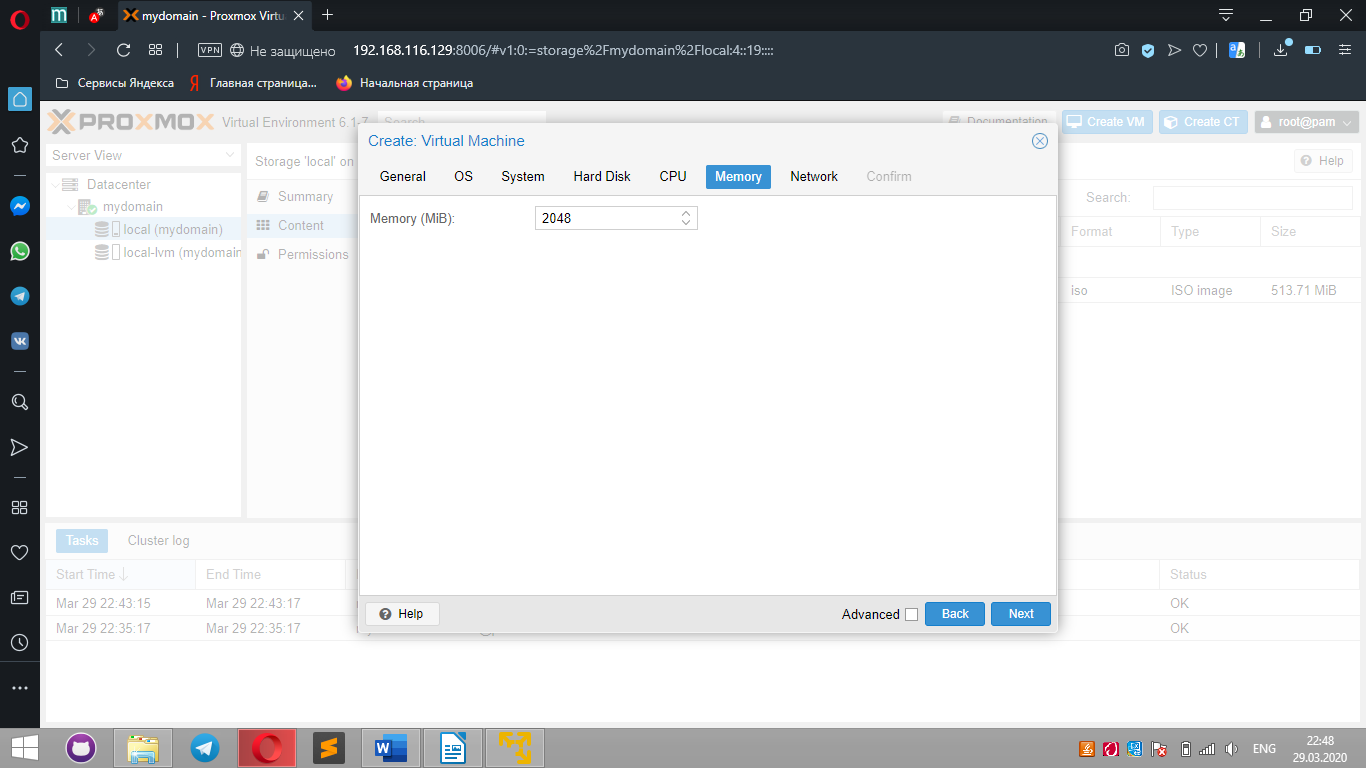
Выбираем размер жесткого диска:



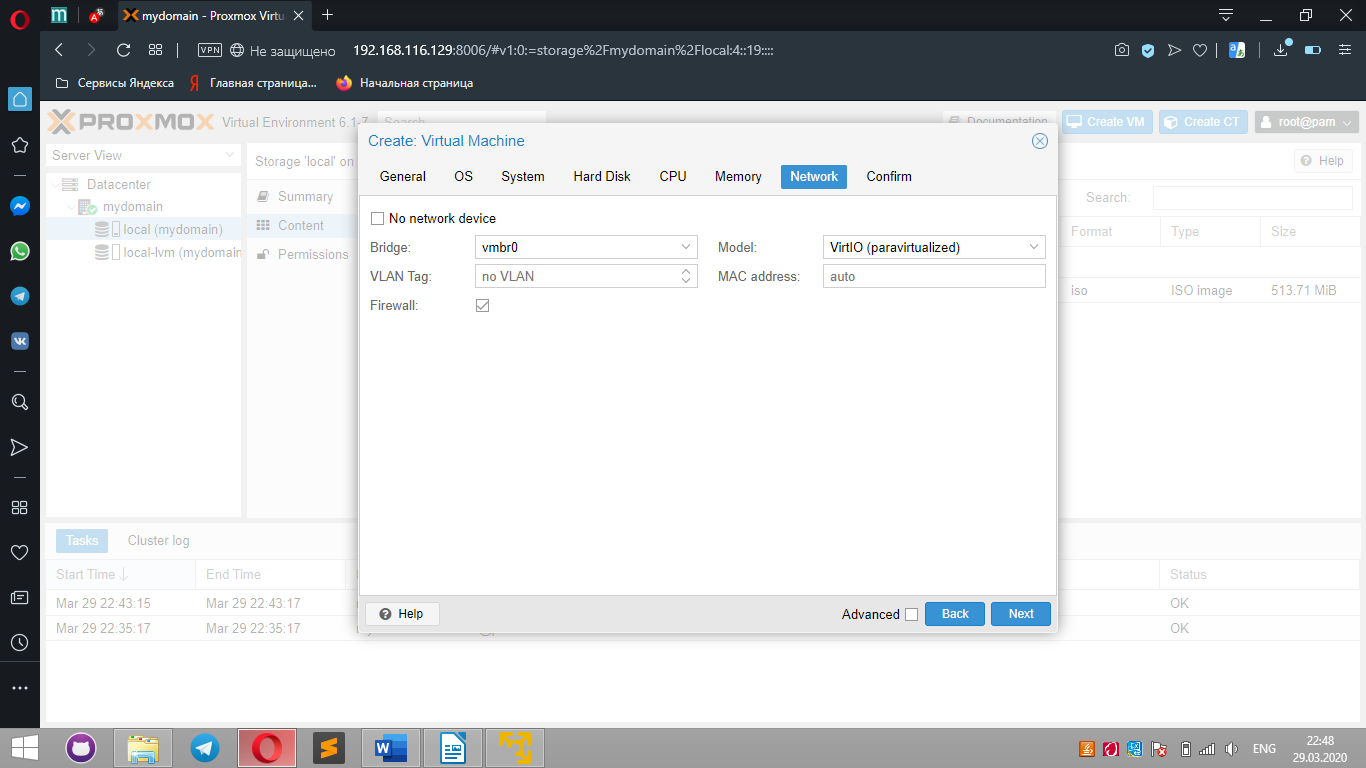
В разделе CPU оставляем всё без изменений:

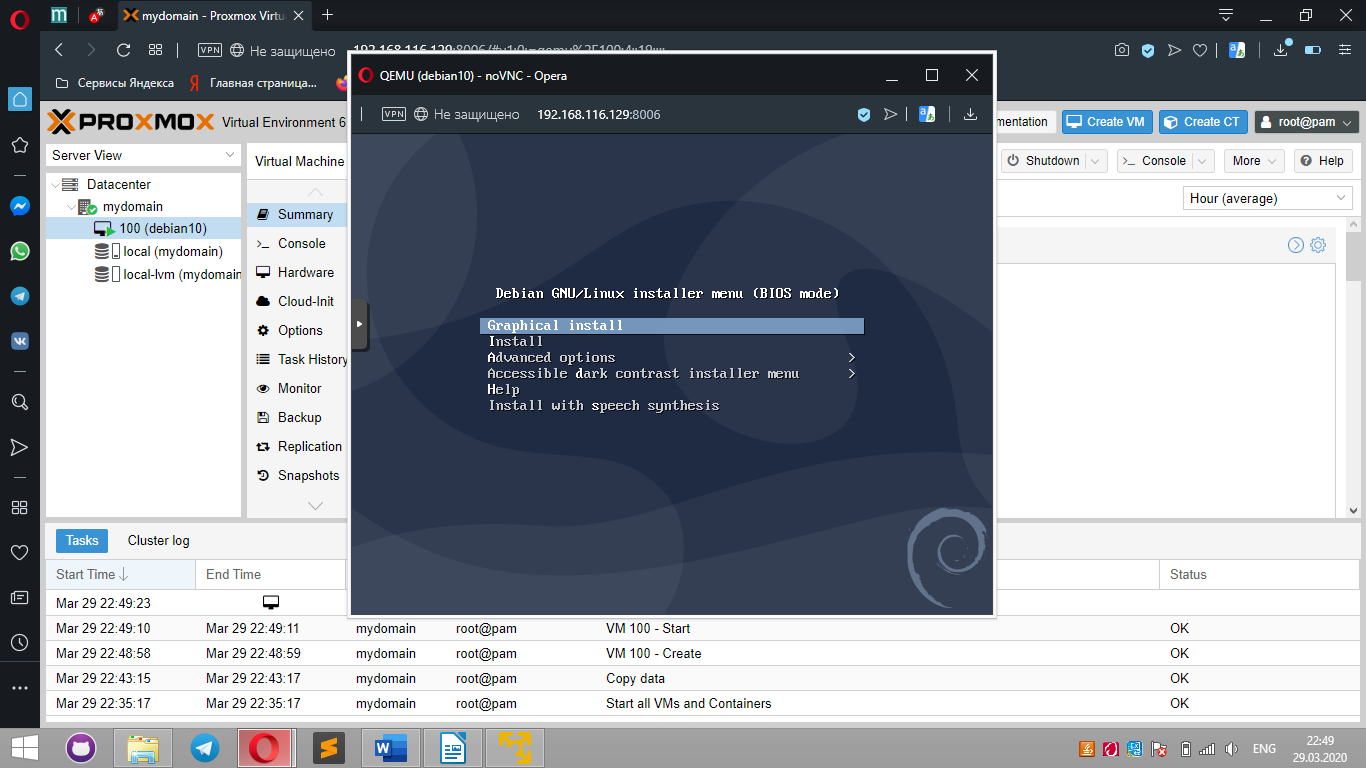


Размер оперативной памяти 2ГБ:

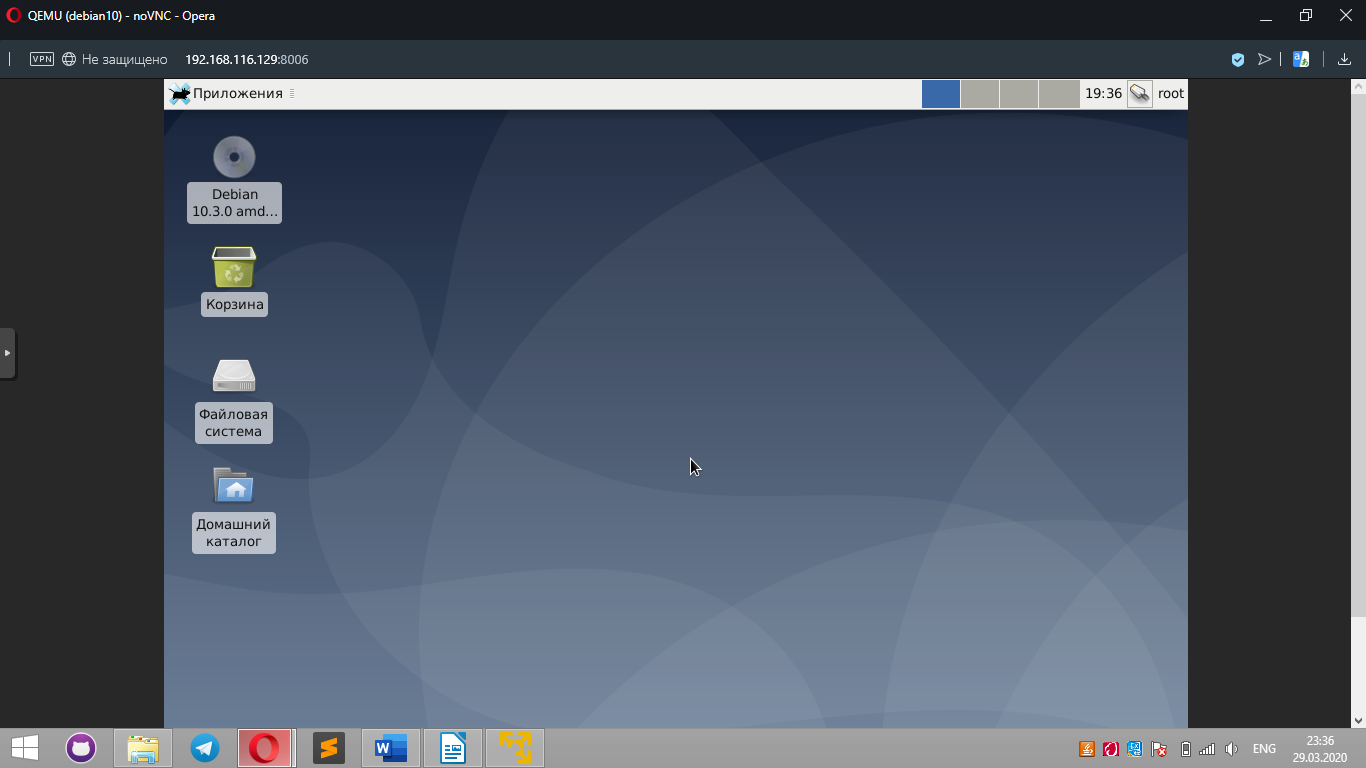


Далее всё тоже без изменений:



Начинаем непосредственную установку системы Debian 10:

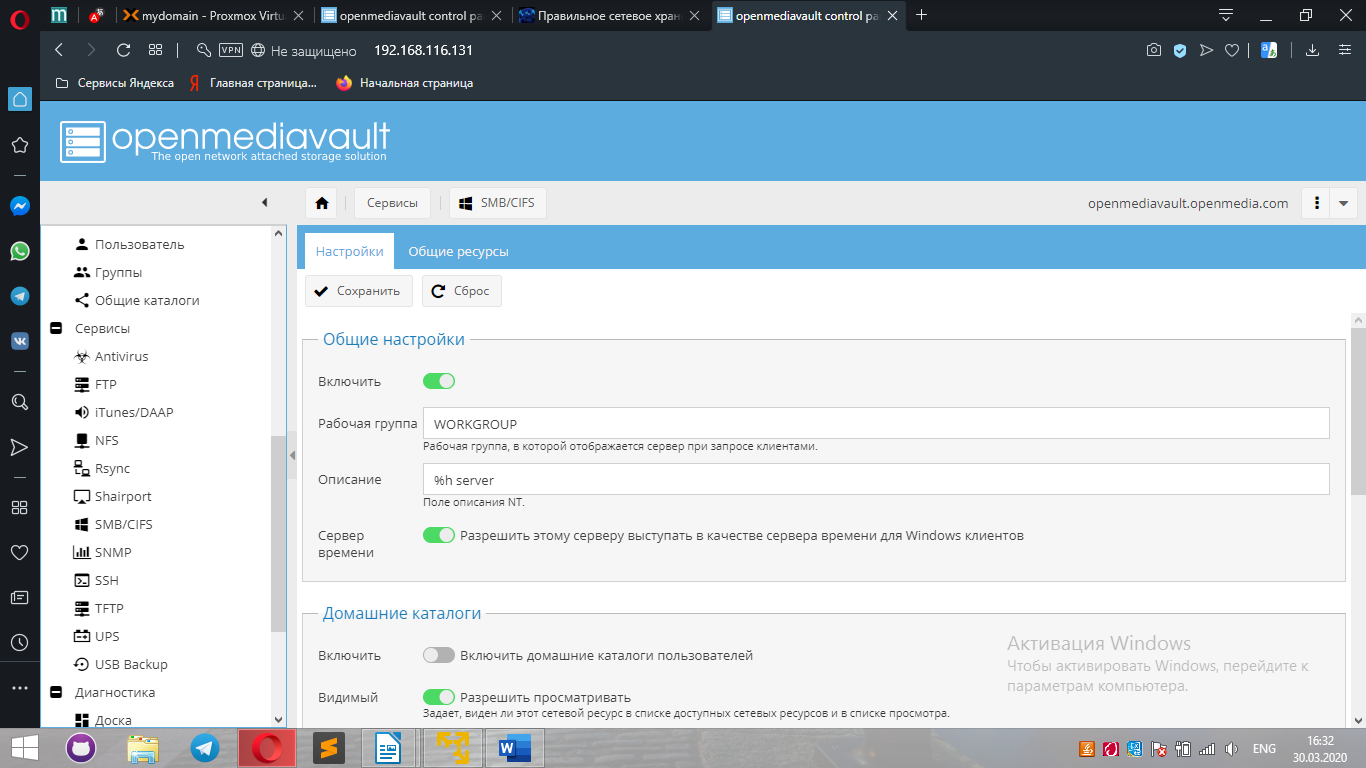
Подробности установки опускаем. Результат работы:



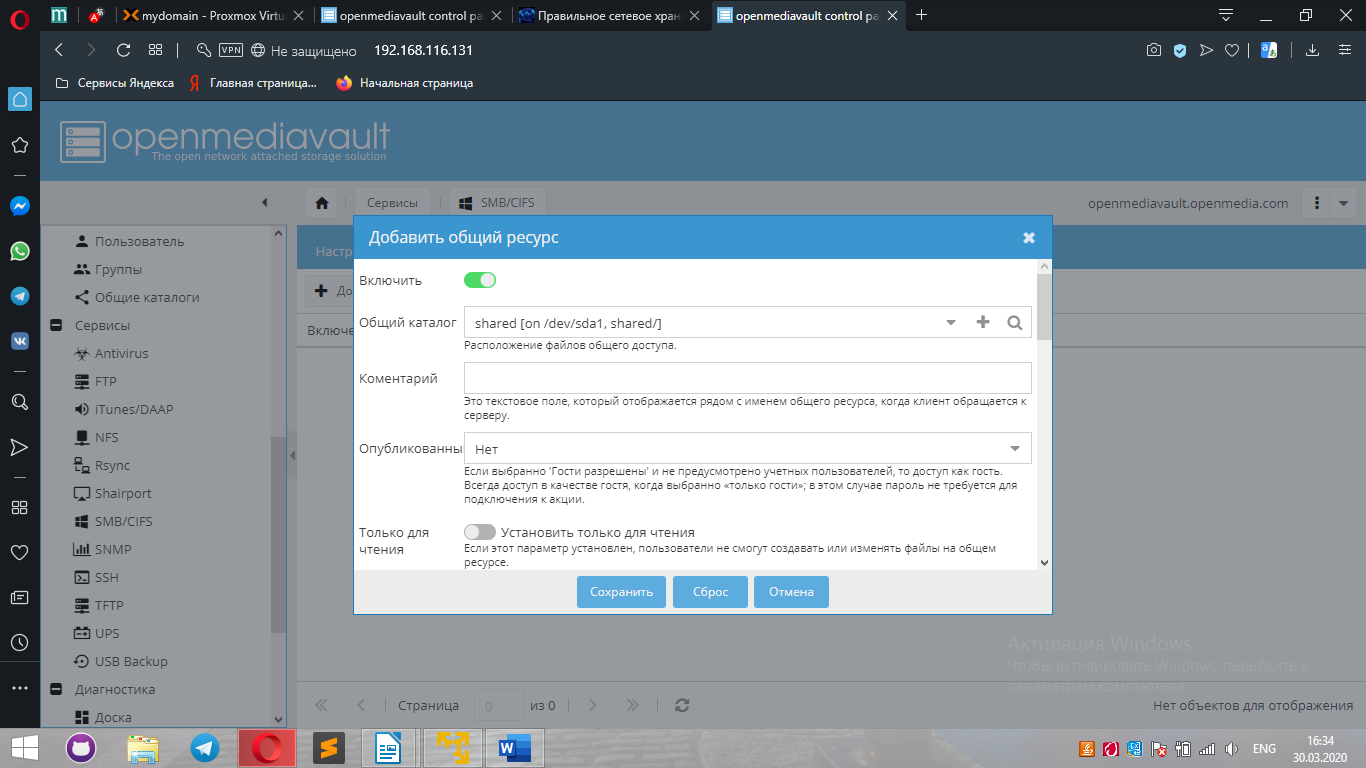
По заданию нужно было подключить хранилку из первой лабораторной. Так как выполнялась на сервере, то была вновь установлена система OpenMediaVault на VMware. Создаём общий ресурс – вводим имя, выбираем устройство, на котором будет общая папка, прописываем путь и устанавливаем права доступа:



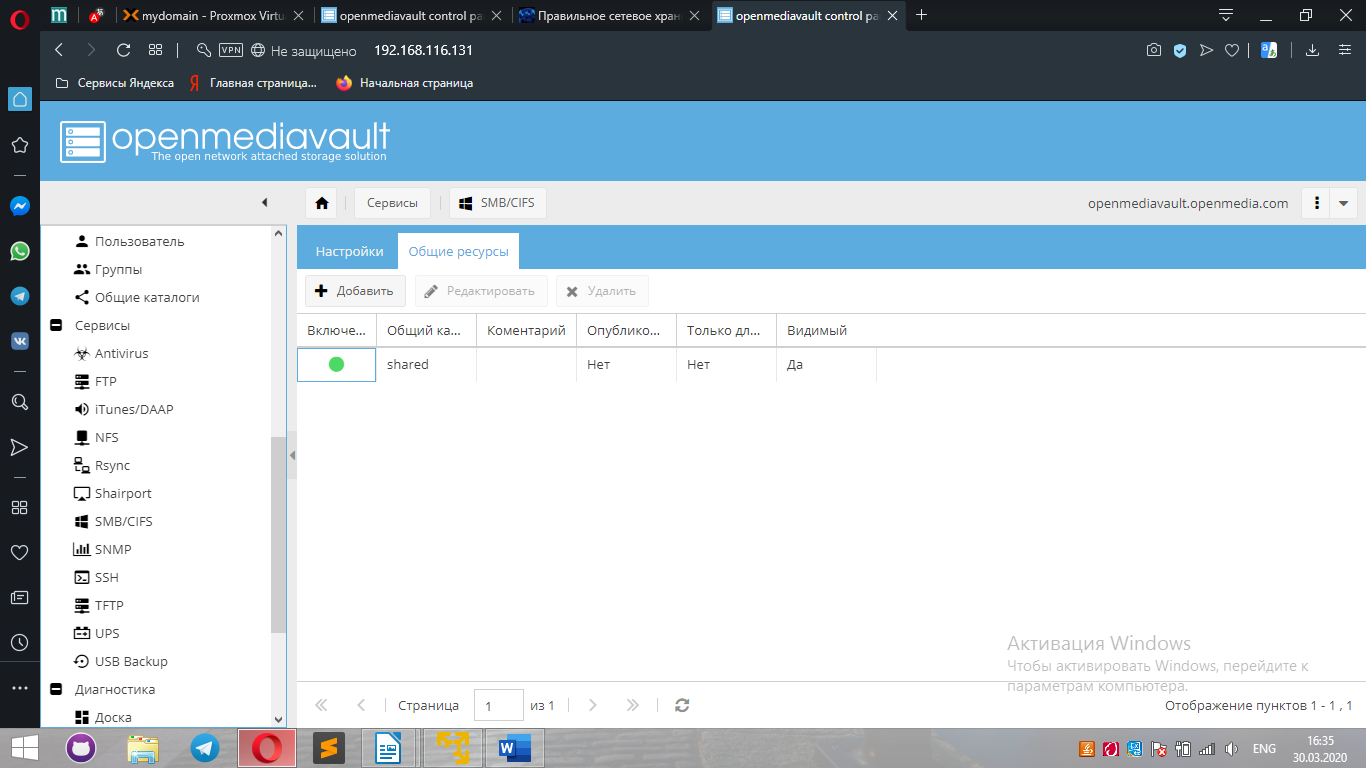
Необходимо открыть доступ к папке через службу Samba:



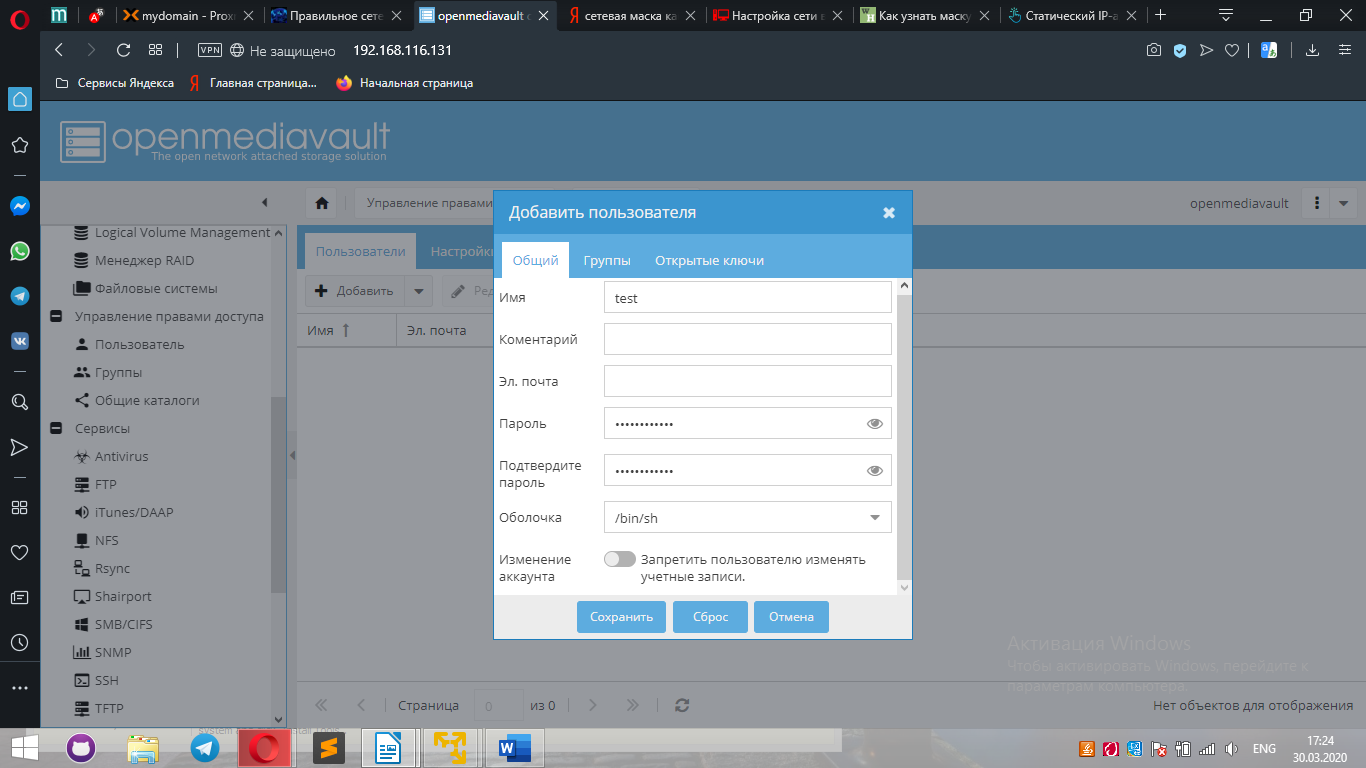
После запуска службы добавляем нашу папку:



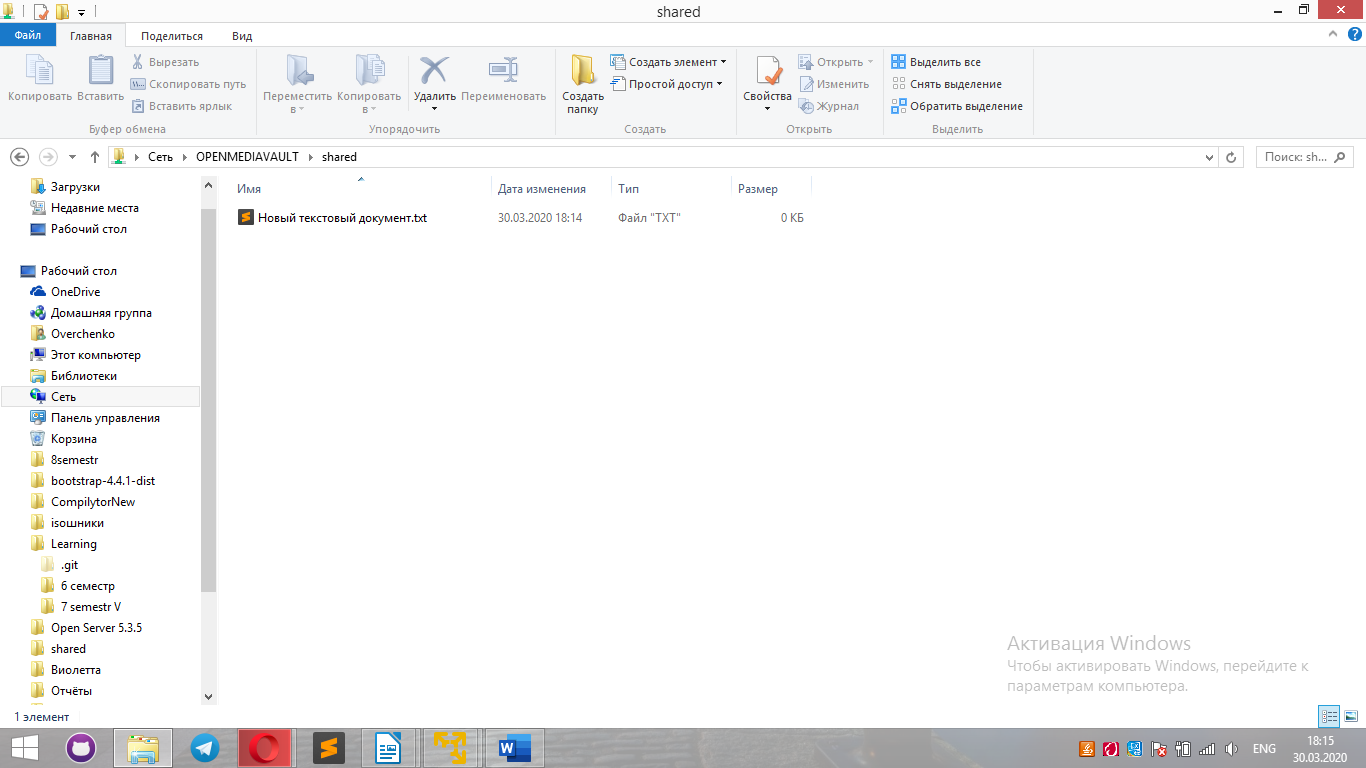
Общая папка стала активной:



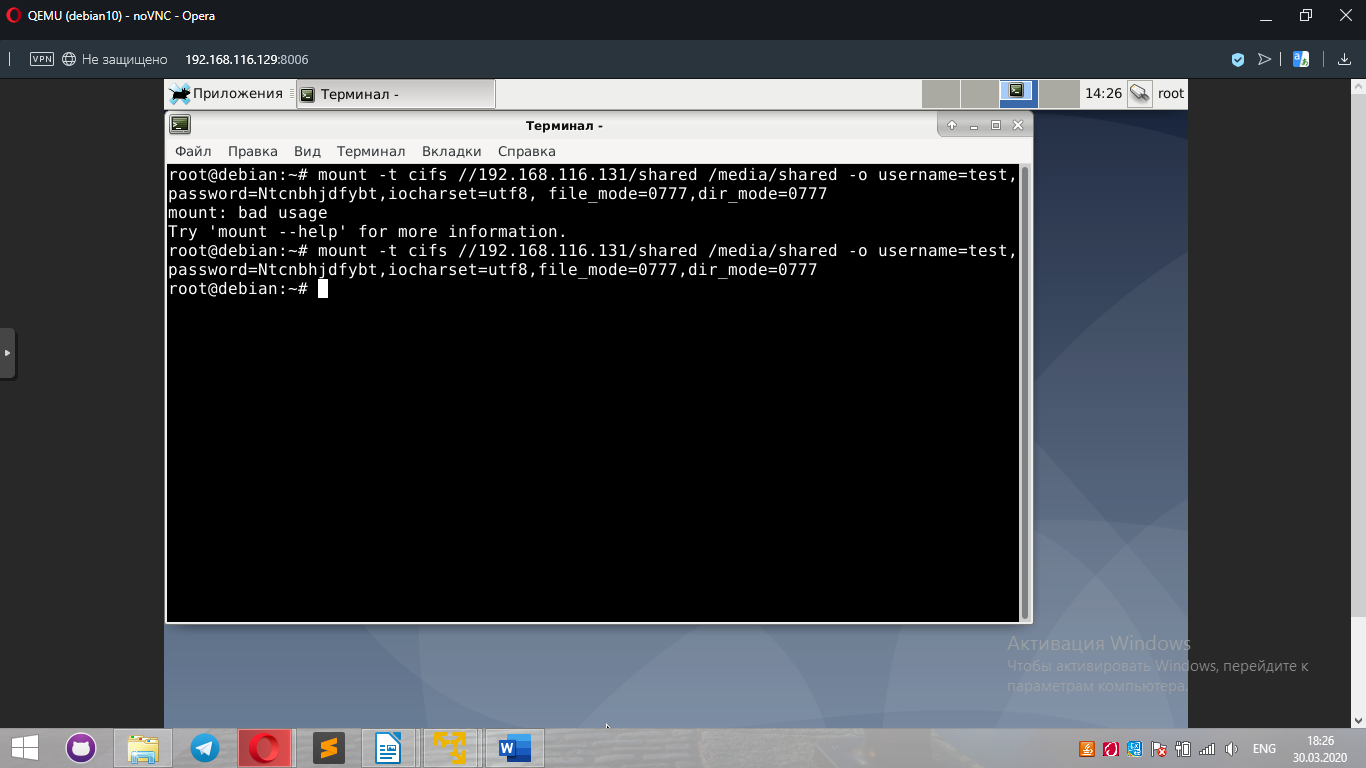
Для проверки работоспособности папки создаем тестового пользователя:



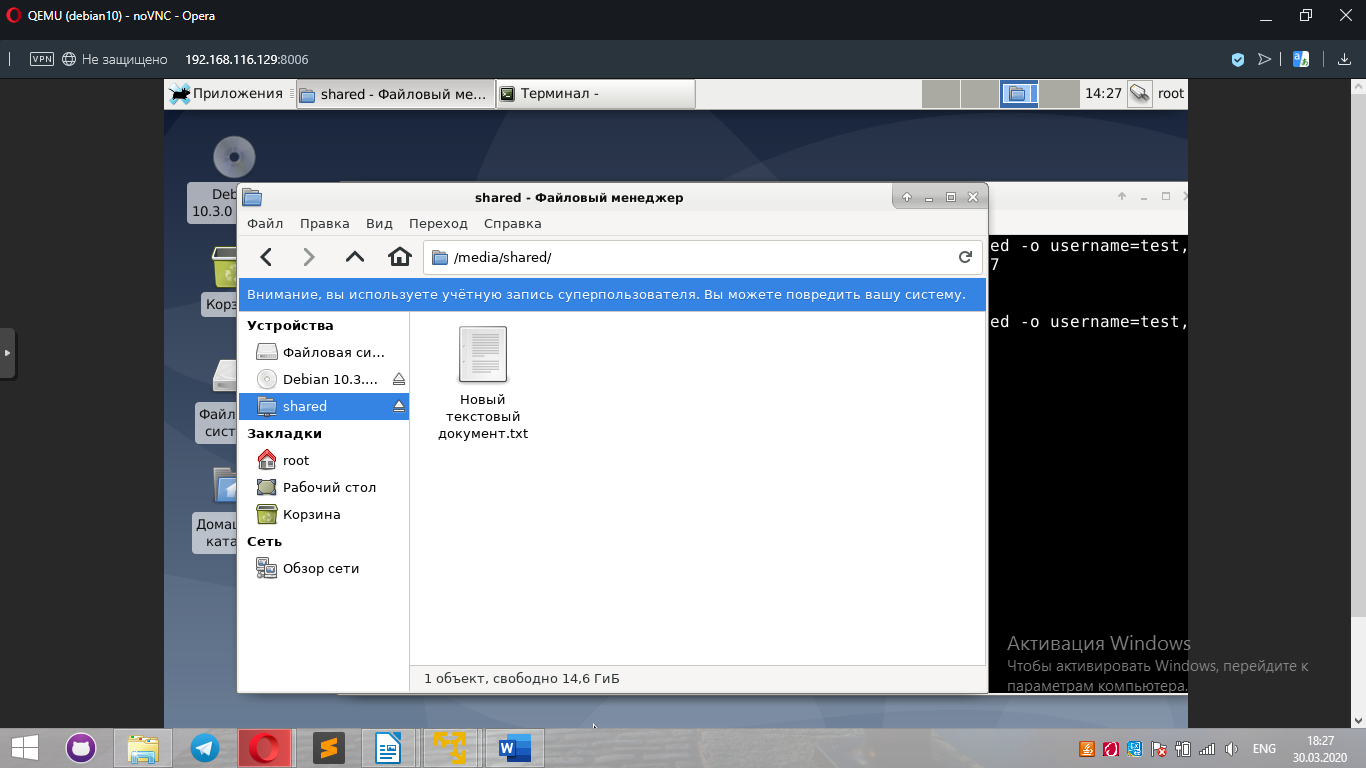
И выполняем вход в общую папку через проводник Windows. Всё работает:



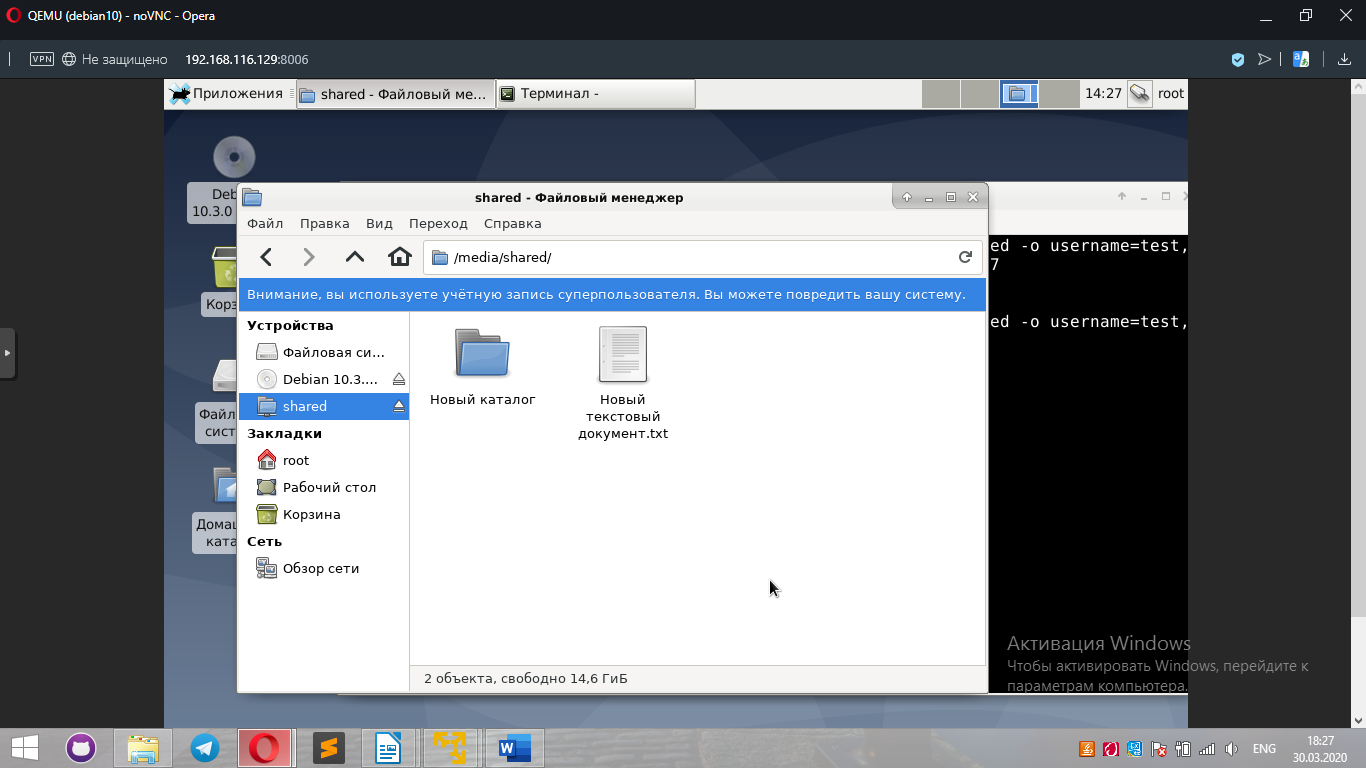
Теперь примонтируем общую папку в системе Debian с помощью соответствующей команды, предварительно установив пакет cifs-utils:



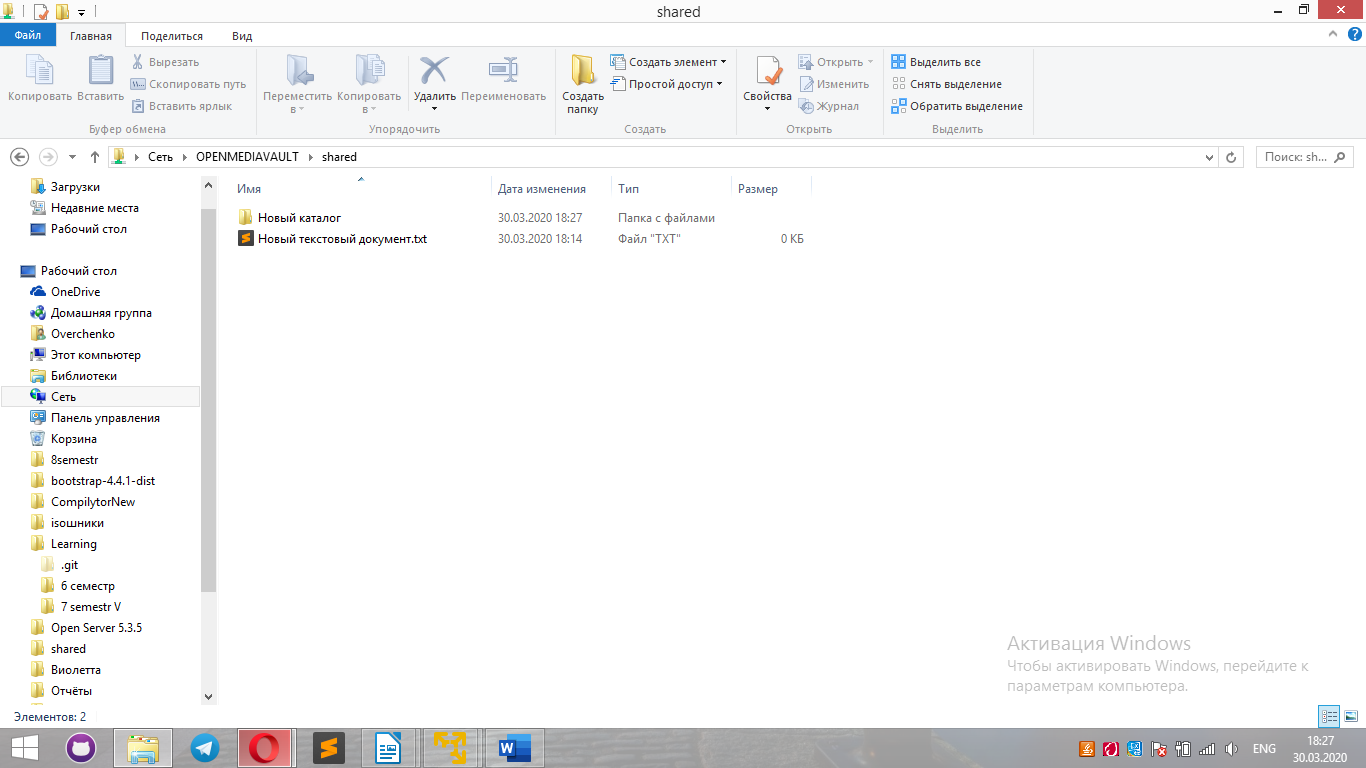
Папка появилась в файловой структуре:



Создадим папку и проверим, что она отображается в среде Windows:



Созданная папка появилась:



Таким образом, видим корректную работу установленной системы Proxmox и подключенной хранилки.

P.S.: Во время создания общего ресурса возникла проблема неизвестного характера. Невозможно было выбрать устройство. Причём такая ситуация была и при создании файловой системы и при создании общей папки. Пробовали инсталлировать заново, читали различные инструкции, перезагружали машину – всё было бесполезно. При очередном включении виртуалки в списке устройств появилось ещё одно и стал доступен выбор (что и заскринено выше).

