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 №4

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

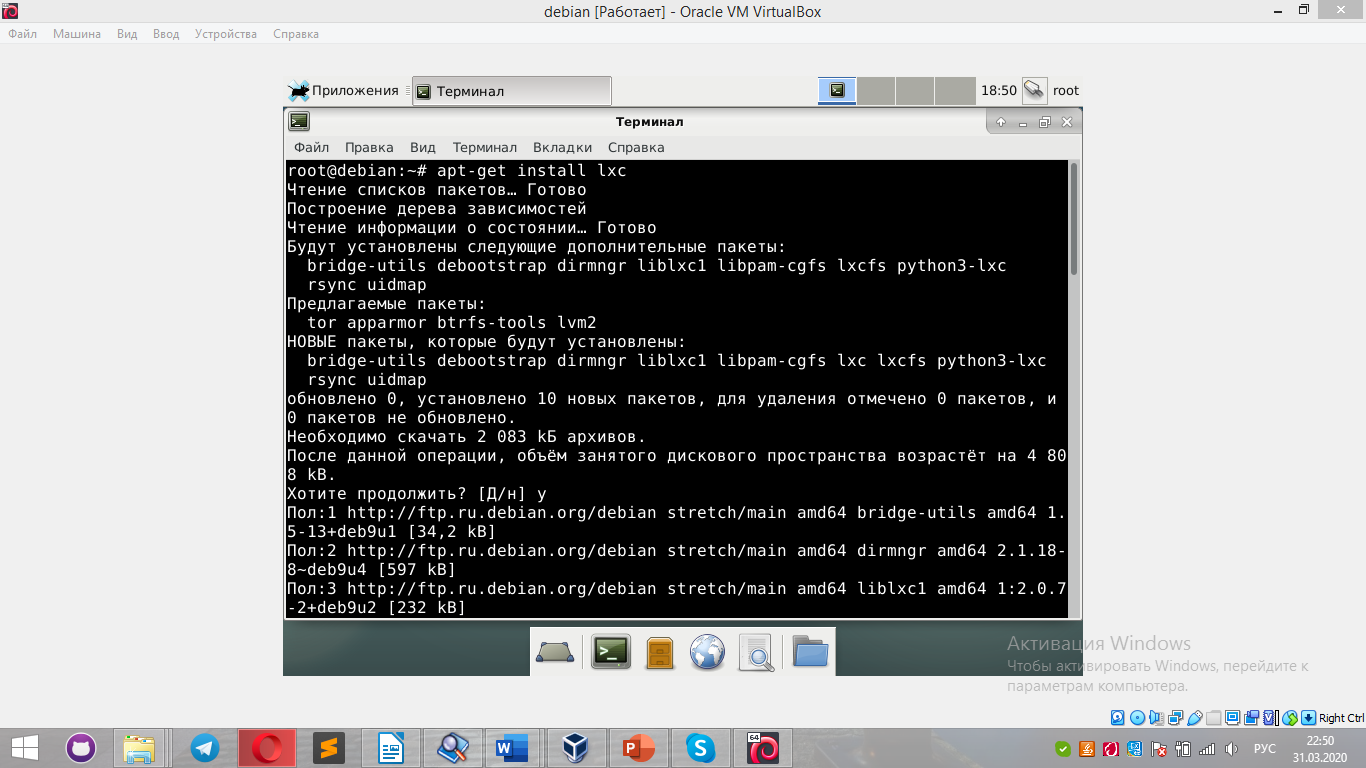
(инициалы, фамилия)

Барнаул 2020

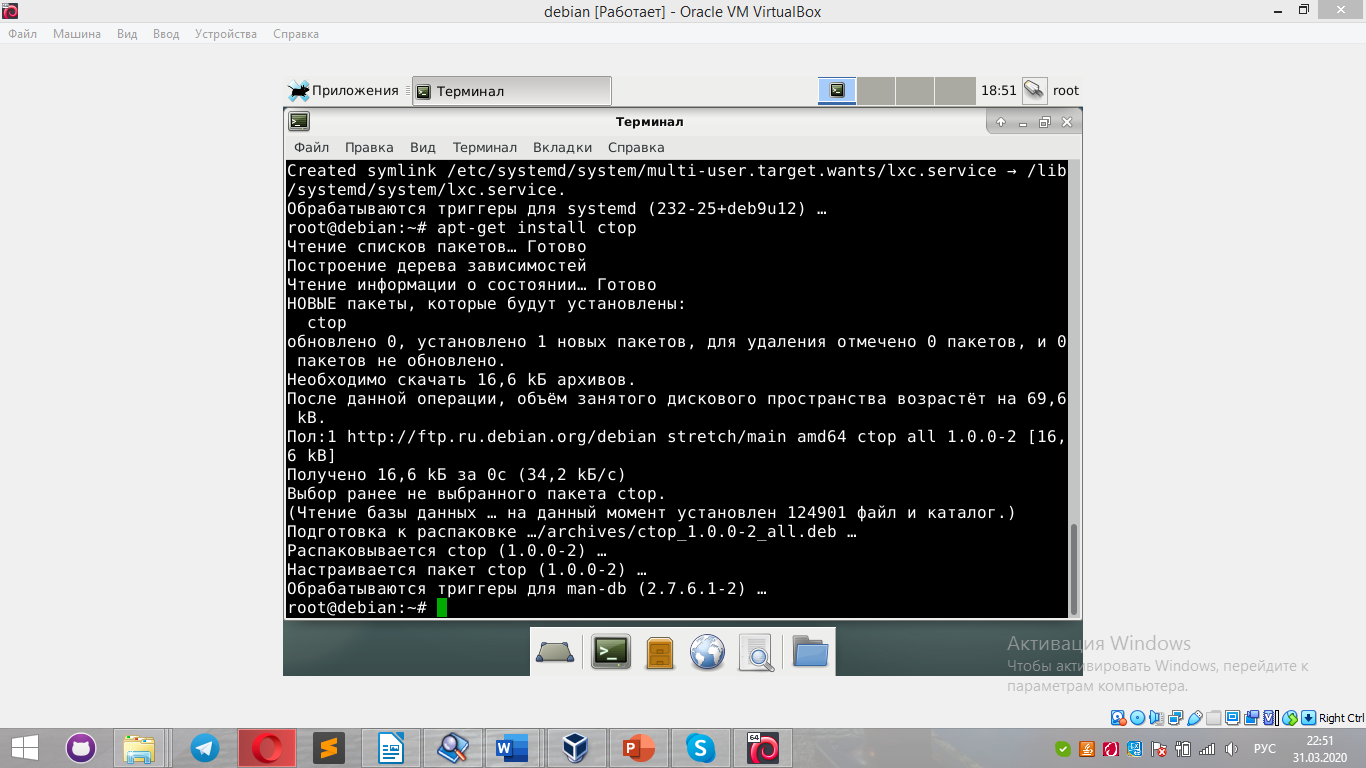
LXC — система виртуализации на уровне операционной системы для запуска нескольких изолированных экземпляров операционной системы Linux на одном узле. LXC не использует виртуальные машины, а создаѐт виртуальное окружение с собственным пространством процессов и сетевым стеком. Все экземпляры LXC используют один экземпляр ядра операционной системы.

Создали виртуальную машину в Proxmox. Установили Debian 9.8.

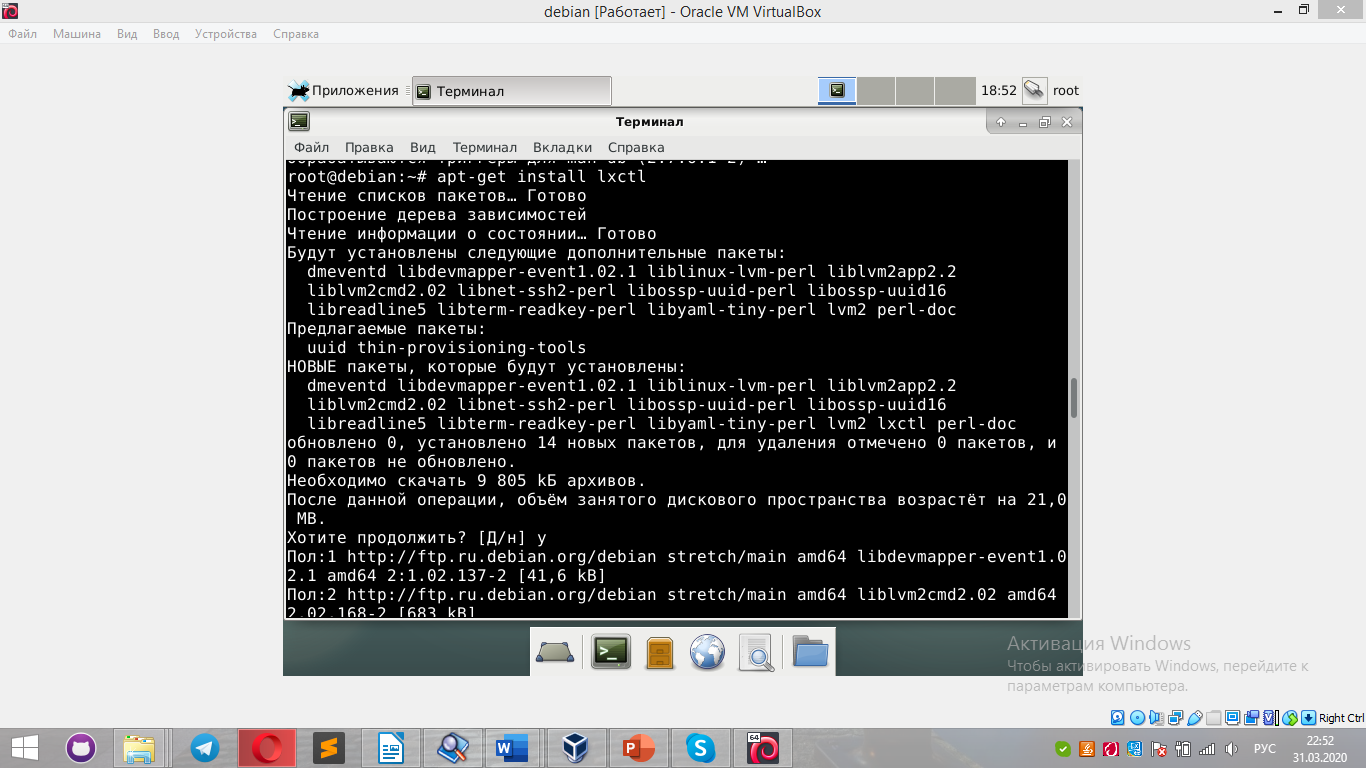
Установили пакет *lxc* и другие зависимые пакеты.



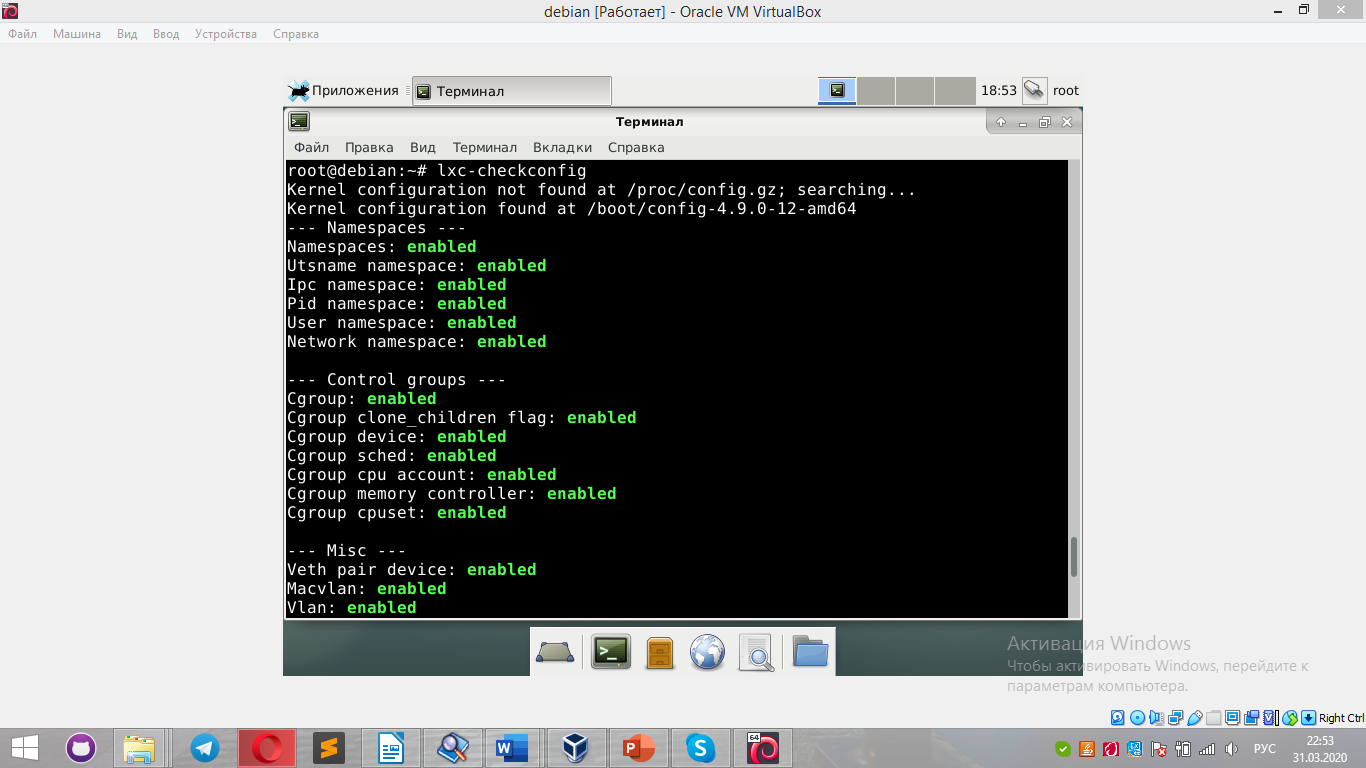
Установили дополнительные пакеты *ctop и lxctl* для наблюдения за работой процессов в контейнерах и управления работой контейнеров.



Lxctl



Проверяем, все ли условия работы LXC выполнены с помощью команды *lxc-checkconfig*

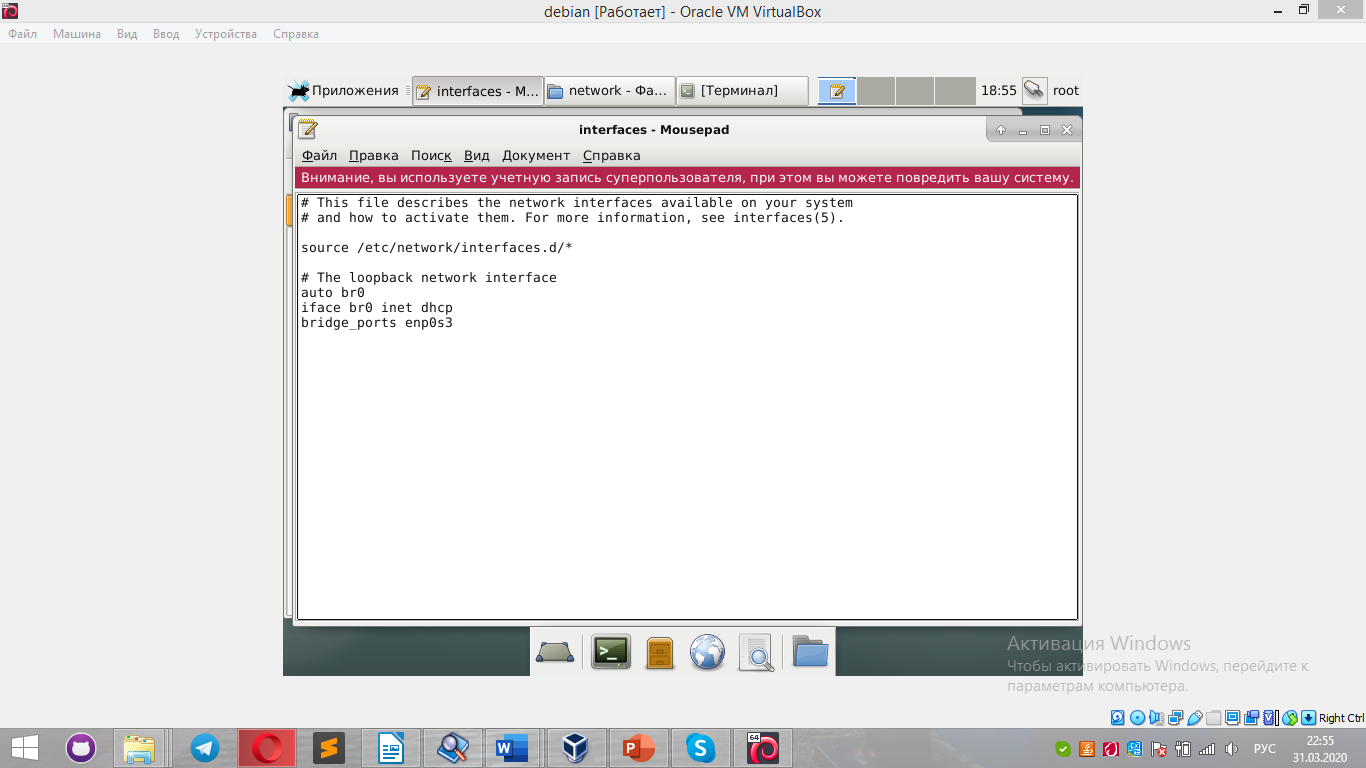


Настраиваем сетевой мост, для этого в файле */etc/network/*interfaces добавляем строки:

auto br0

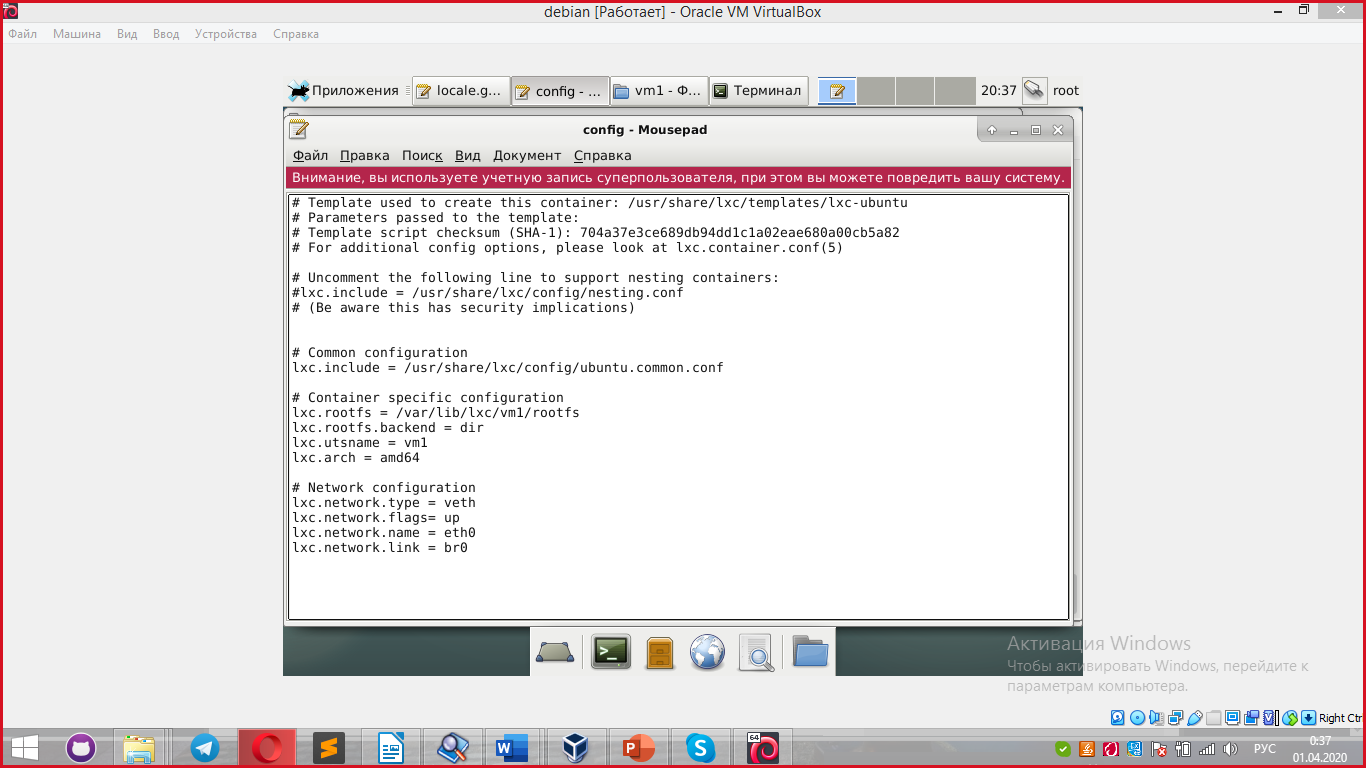
interface br0 inet dhcp

bridge\_ports enp0s3

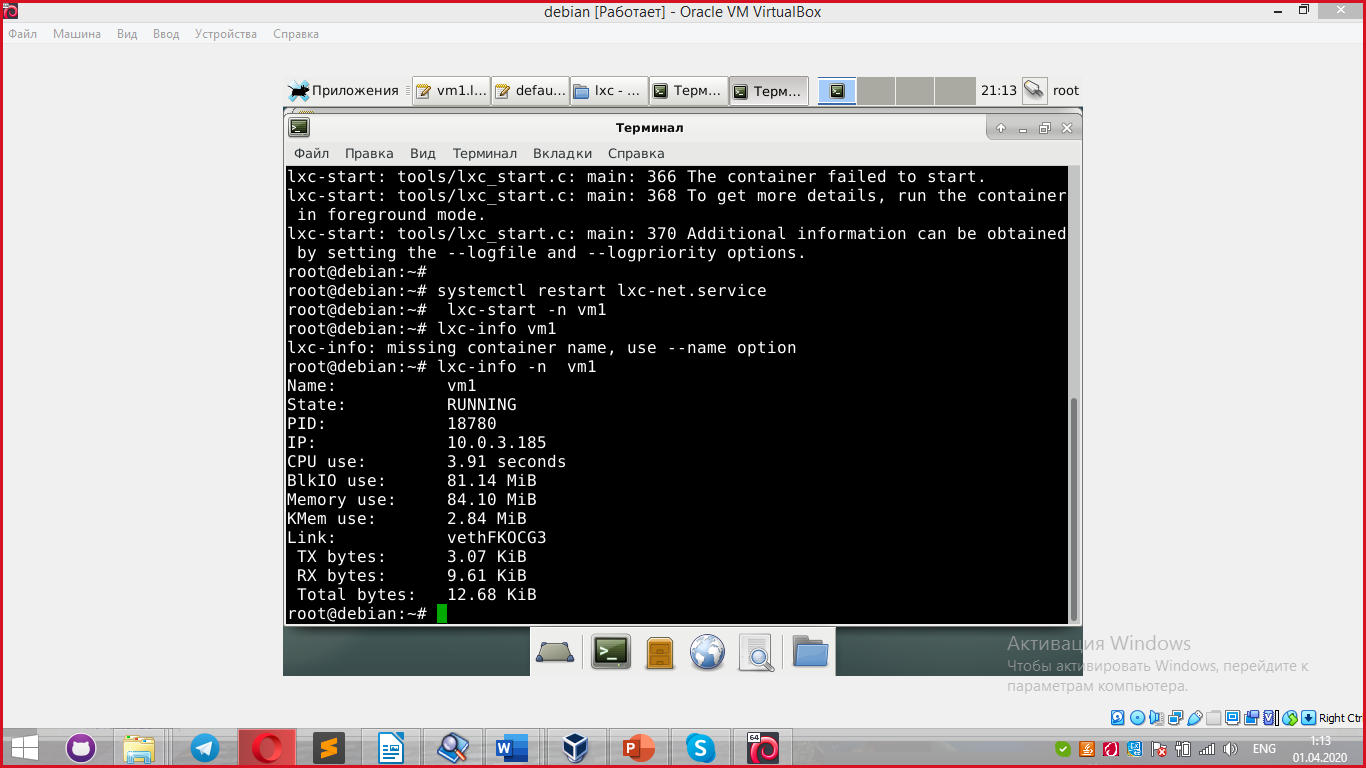


Создали контейнер **vm1** с операционной средой ubuntu (к сожалению, скрин не сохранился, но ниже будет видны данные о контейнере).

Подправим файл конфигурации для нормального функционирования сети:

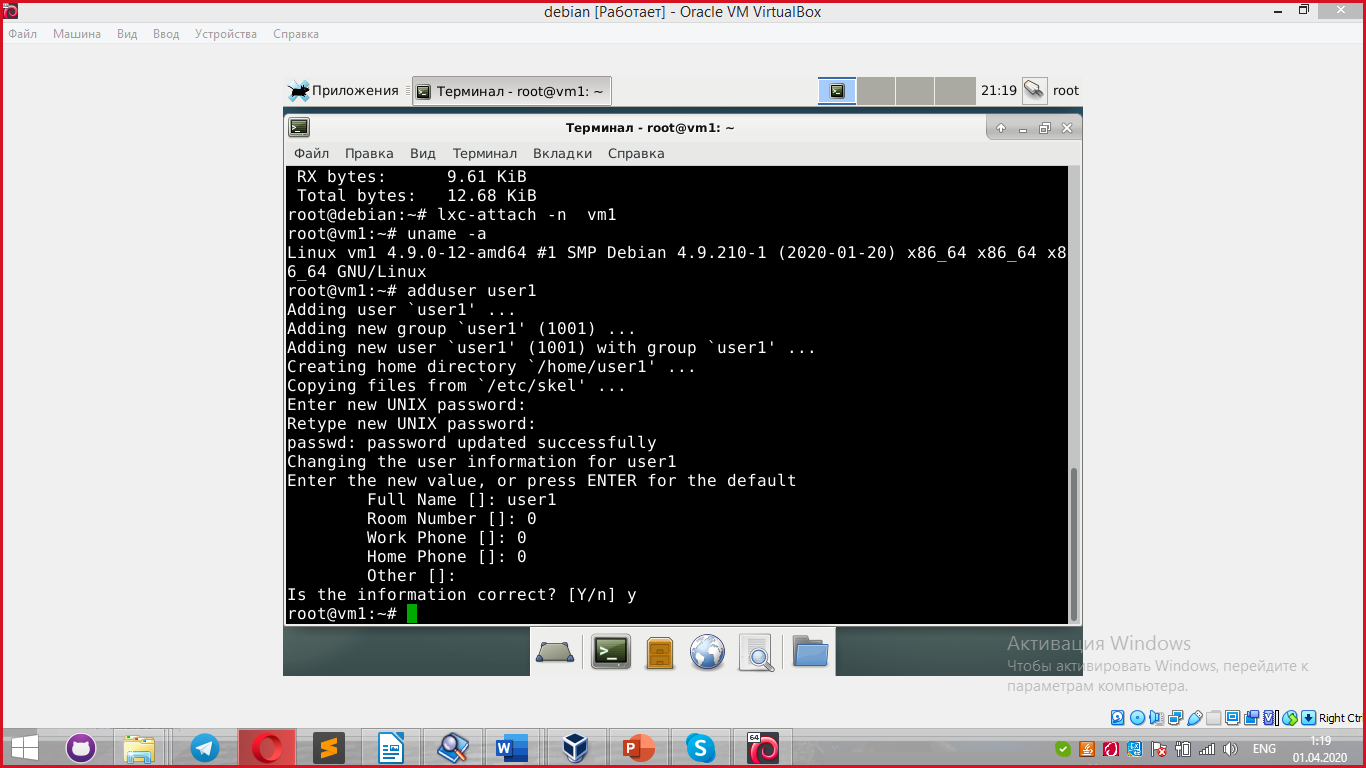


Запустим виртуальную машину с помощью команды lxc-start –n vml, команда lxc-ls показывает все контейнеры, а команда lxc-info — информацию о конкретной виртуальной машине vml.

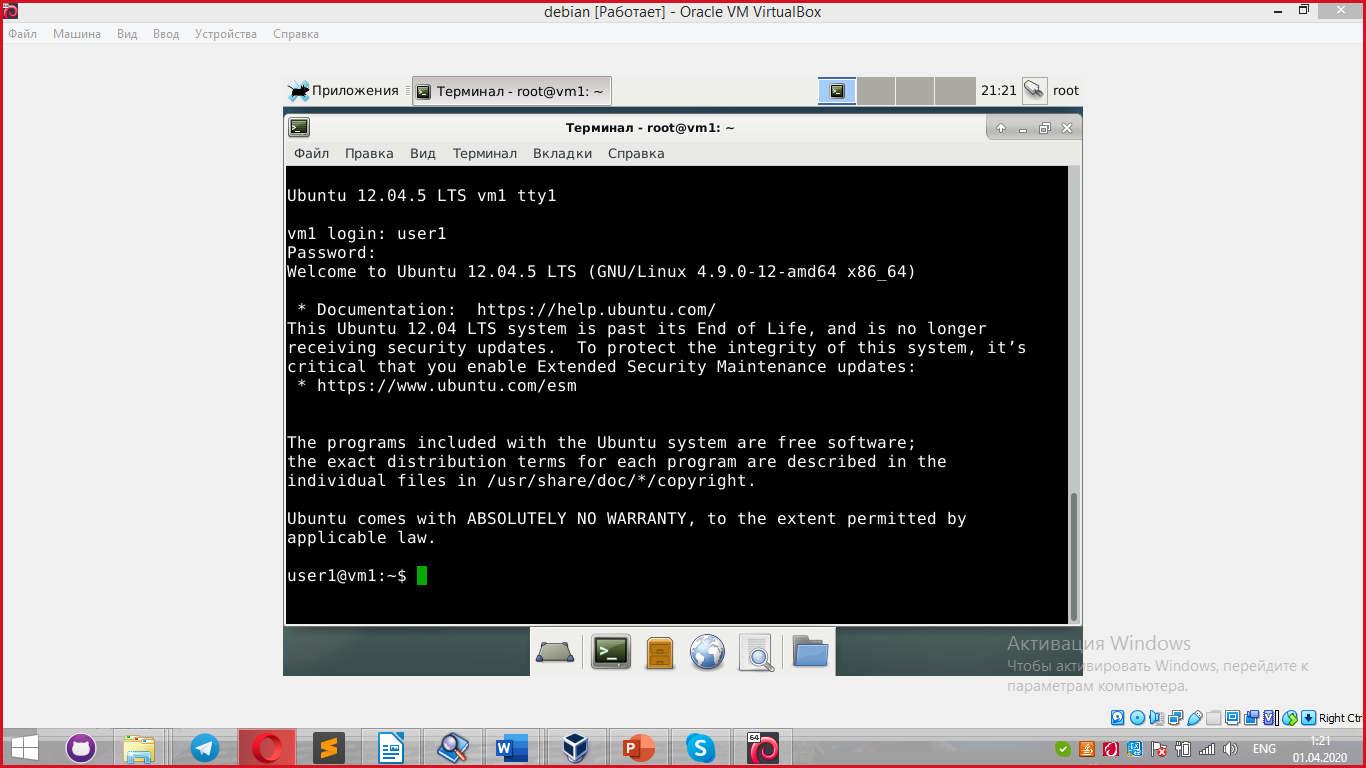


Создадим нового пользователя user1 и установим ему пароль.

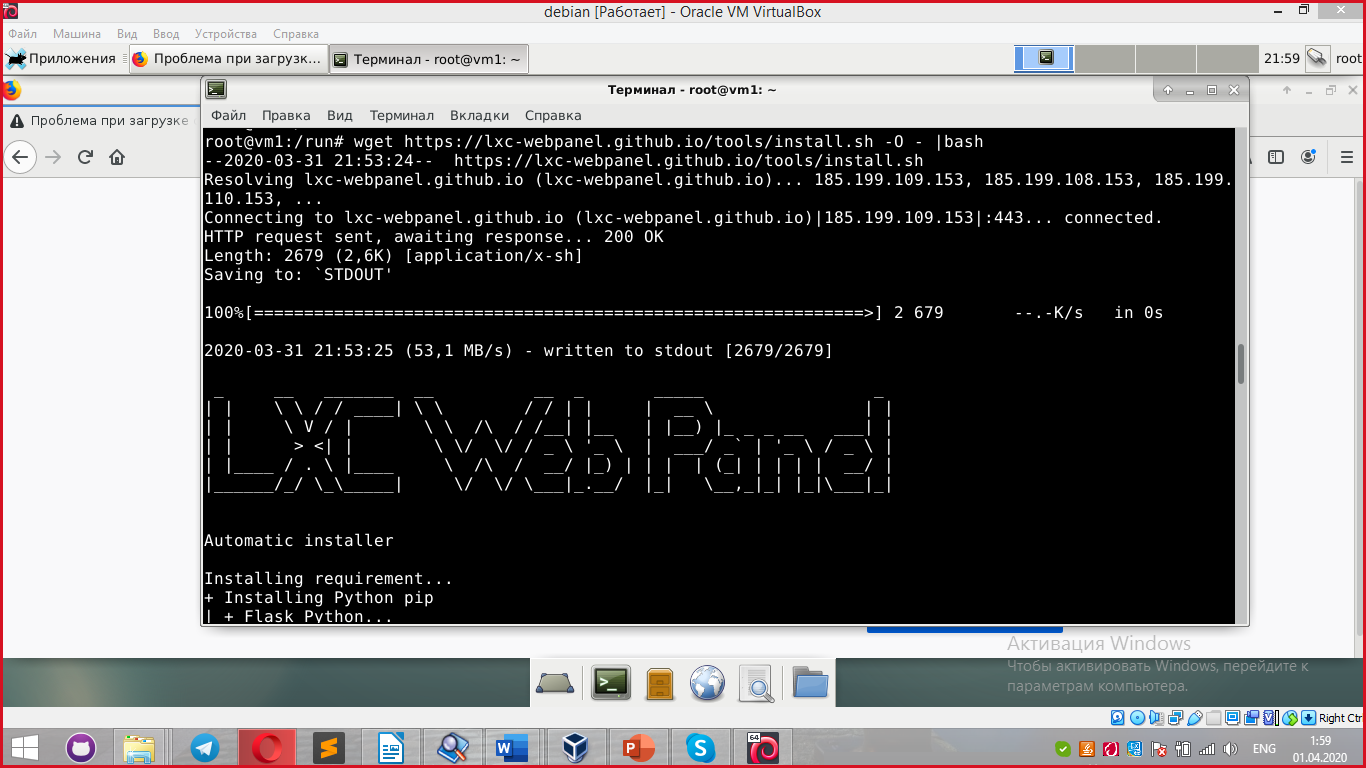
Заполняем необходимую информацию о нем.



Авторизовались под ним



C помощью команды *wget https://lxc-webpanel.github.io/tools/install.sh -O - |bash* запустим скачивание и установку LXC Web Panel – это web-управление LXC-контейнерами.



После окончания инсталляции безуспешно пытались перейти по адресу localhost:5000.

Далее предполагали и испробовали массу способов добиться желаемого. Пробовали также на debian 10, установленном в Proxmox.

Устанавливали python 3.6, всячески конфигурировали файлы настройки сети.

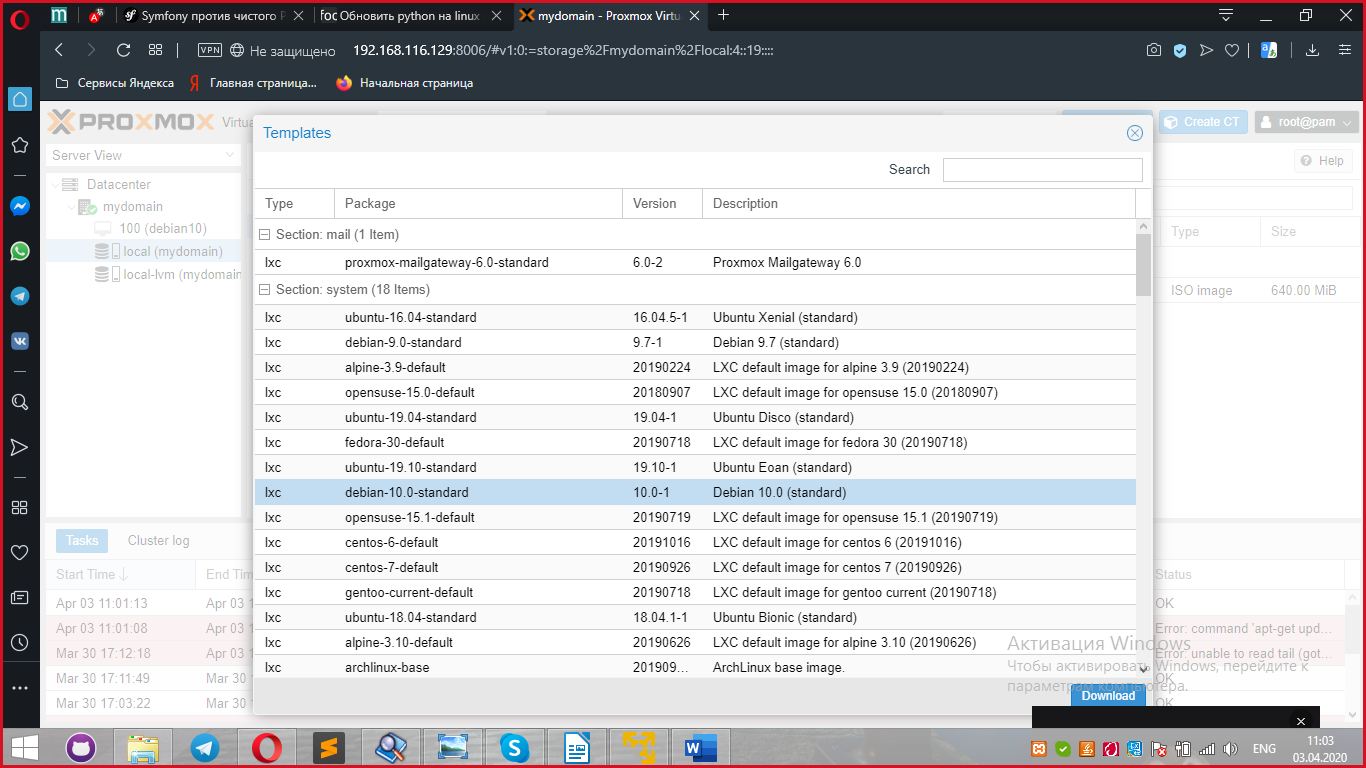
Т.к. netstat не отображал порт 5000, предположили, что проблема в том, что он, как оказалось, был закрыт. Открыли на хосте, делали проброс портов в Virtual box. Попытались порт открыть в гостевой системе, вроде без ошибок, но порт не отображается в прослушиваемых.

**Proxmox**

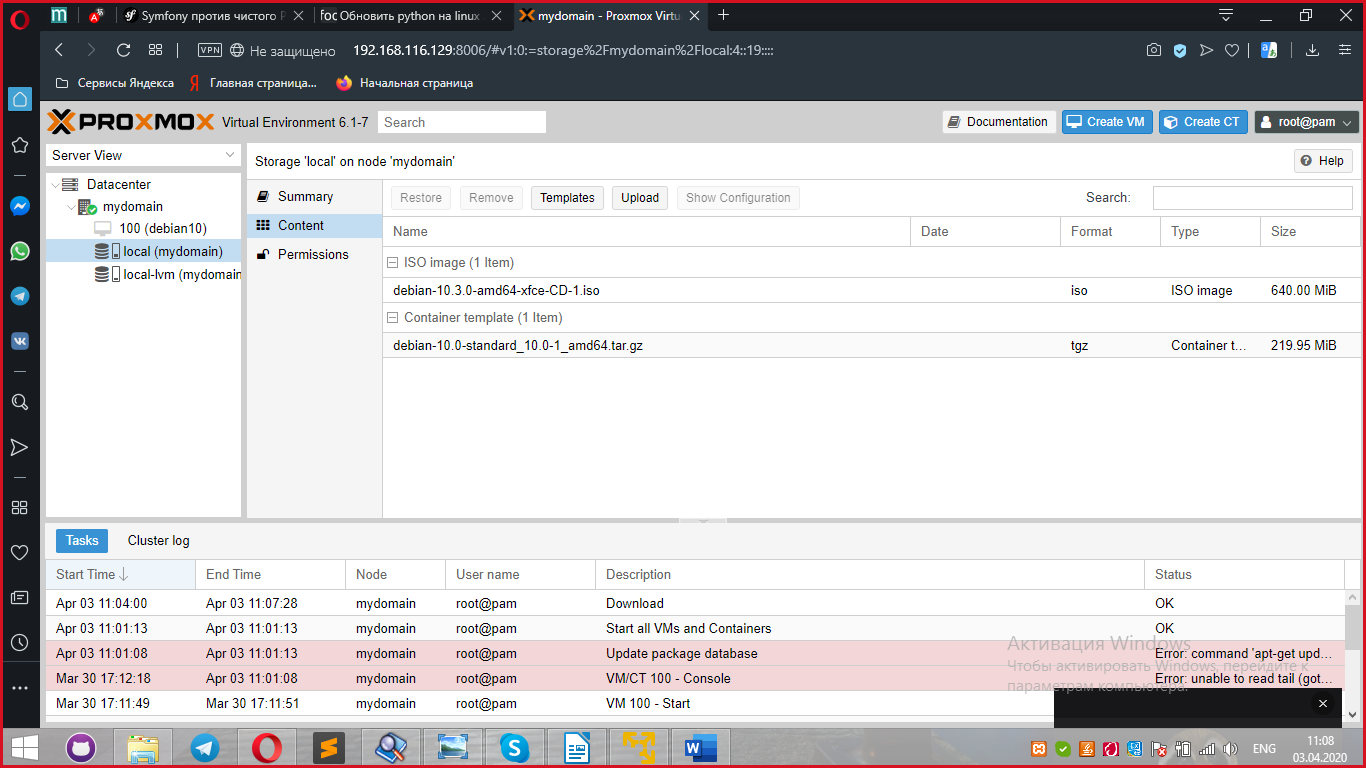
Шаблон является основой для контейнера LXC (в Proxmox VE принято сокращение "CT"). Существуют два сорта шаблонов: шаблоны ОС и шаблоны устройств.

Шаблоны операционной системы предоставляют минимальный набор пакетов из дистрибутива GNU/ Linux а также системных библиотек и сценариев для запуска и работы в контейнере. Они разработаны для полного извлечения преимуществ от работающего на вашем хосте ядра GNU/ Linux.

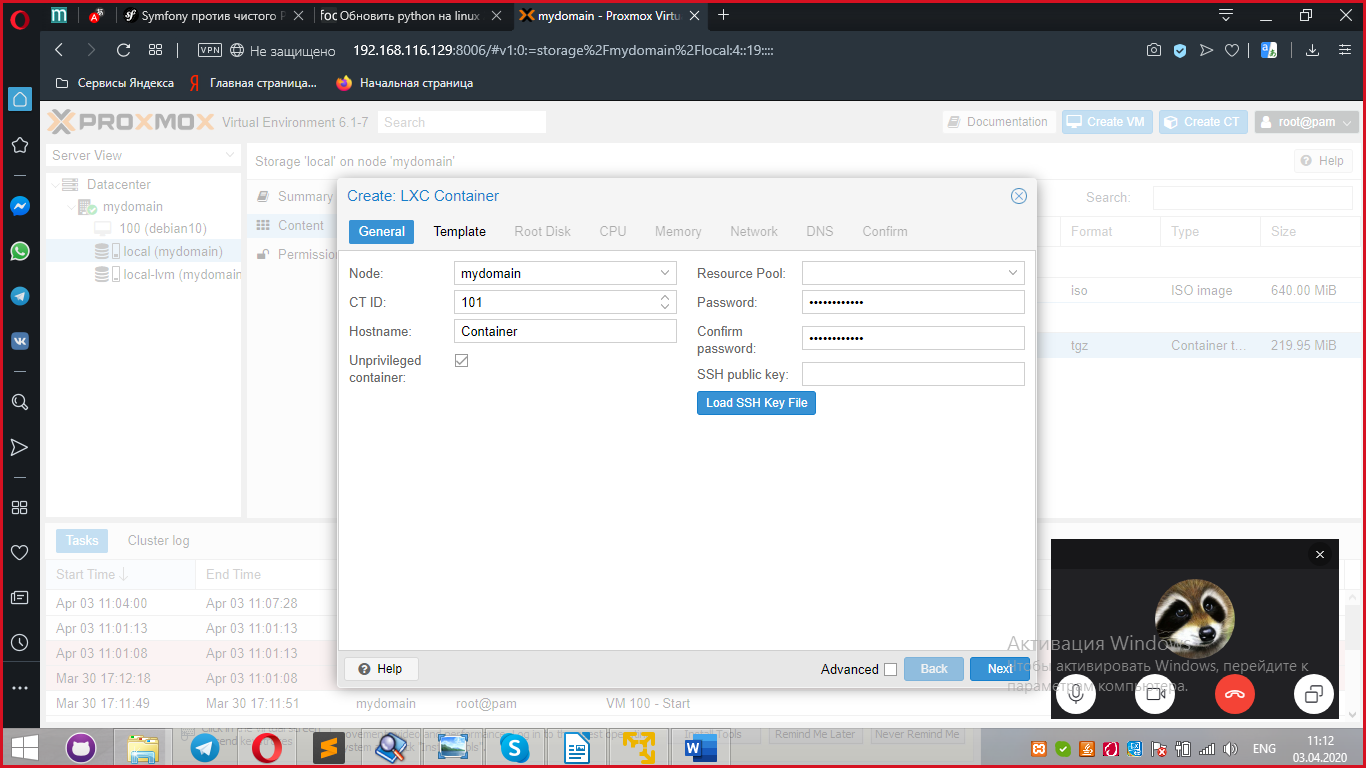
Выбираем шаблон debian 10 и загружаем:



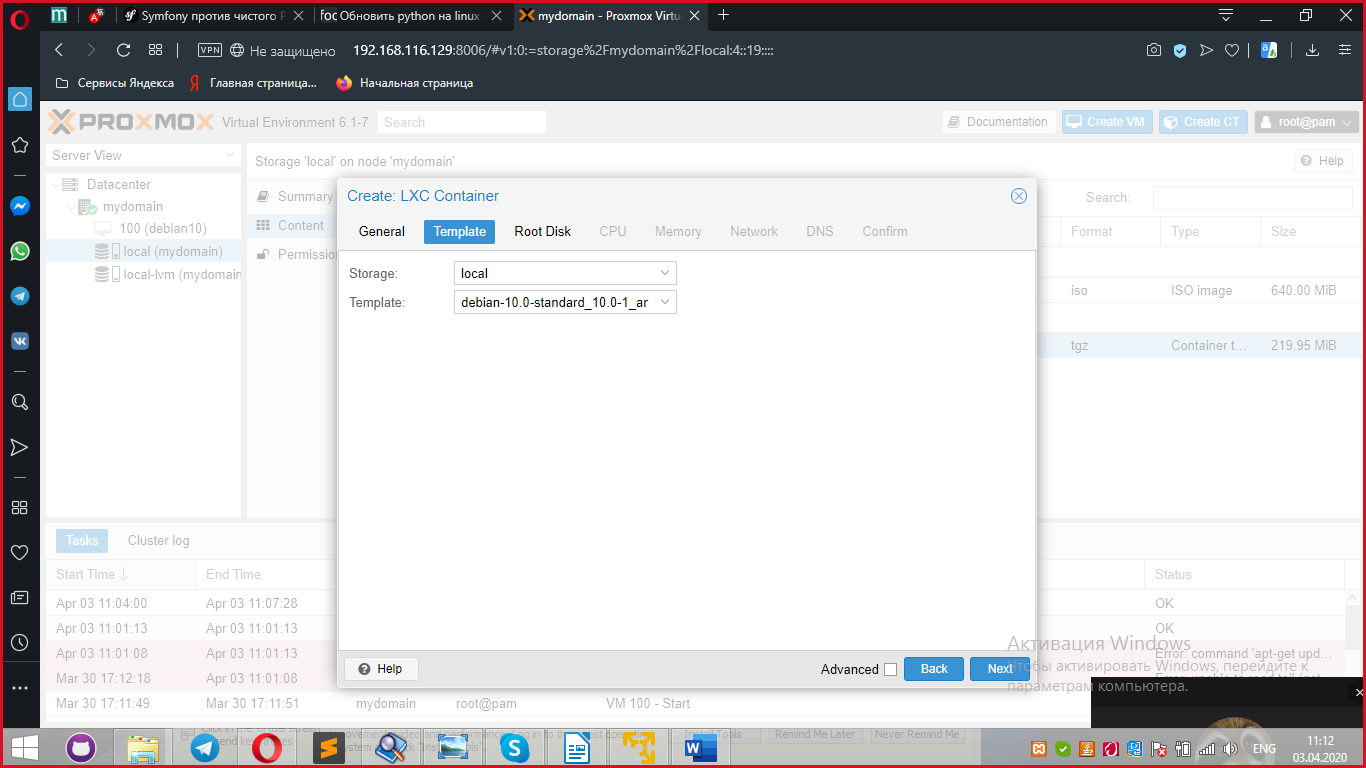
Шаблон загрузился в хранилище:



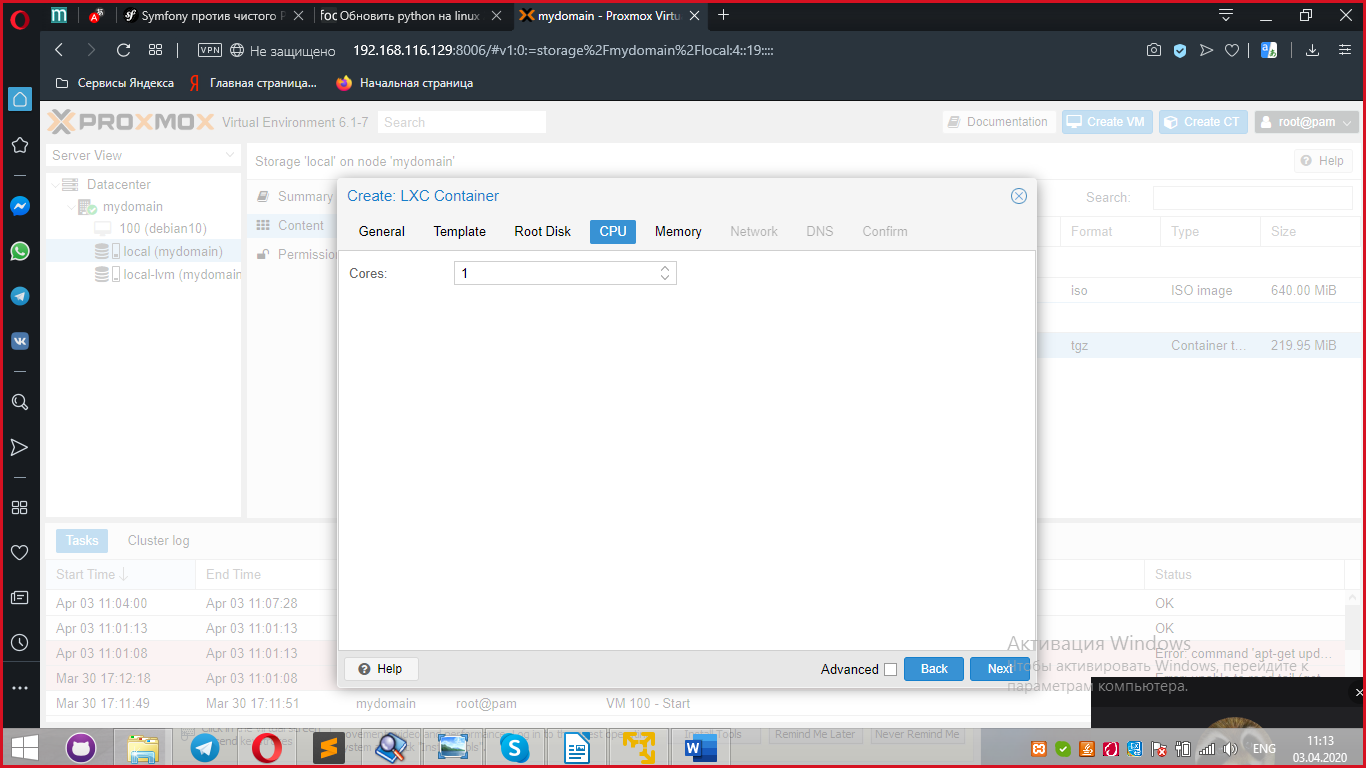
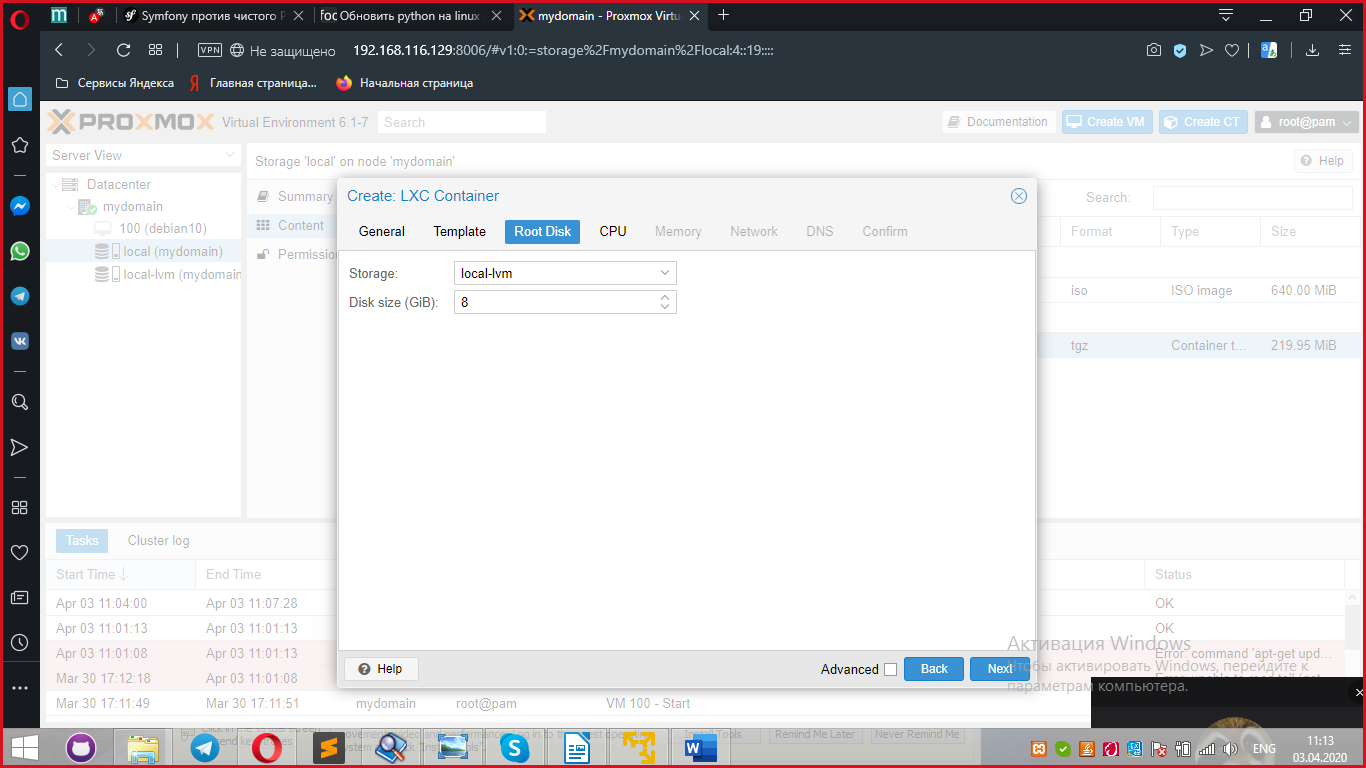
Создаем контейнер.

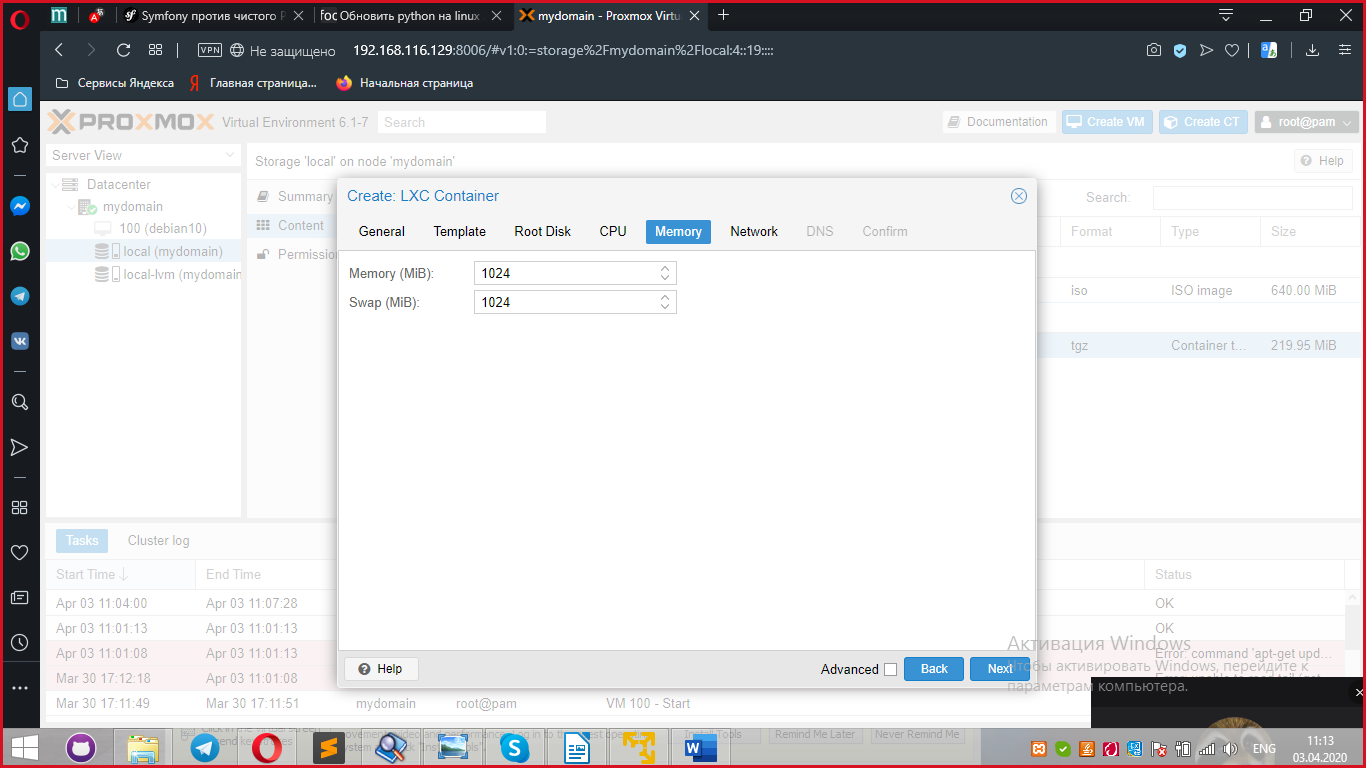


Выбираем шаблон контейнера:

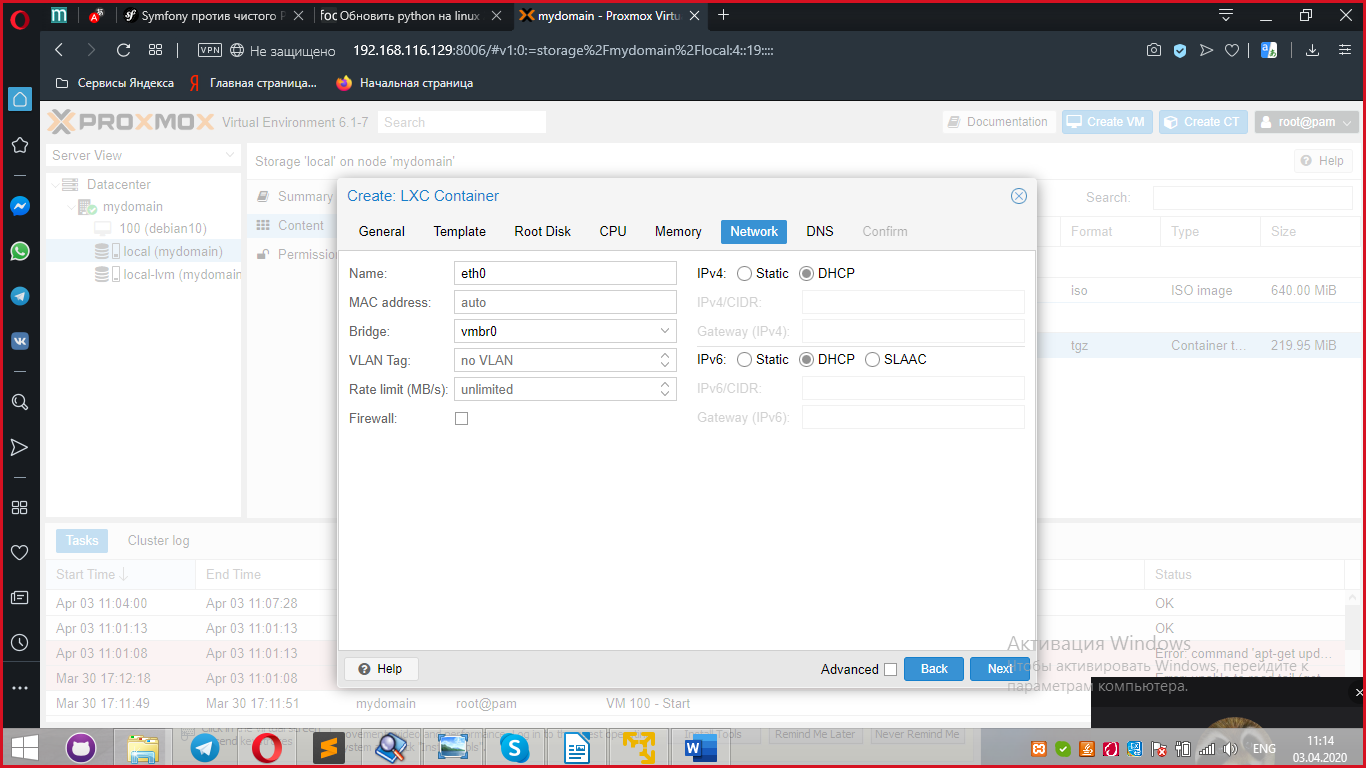


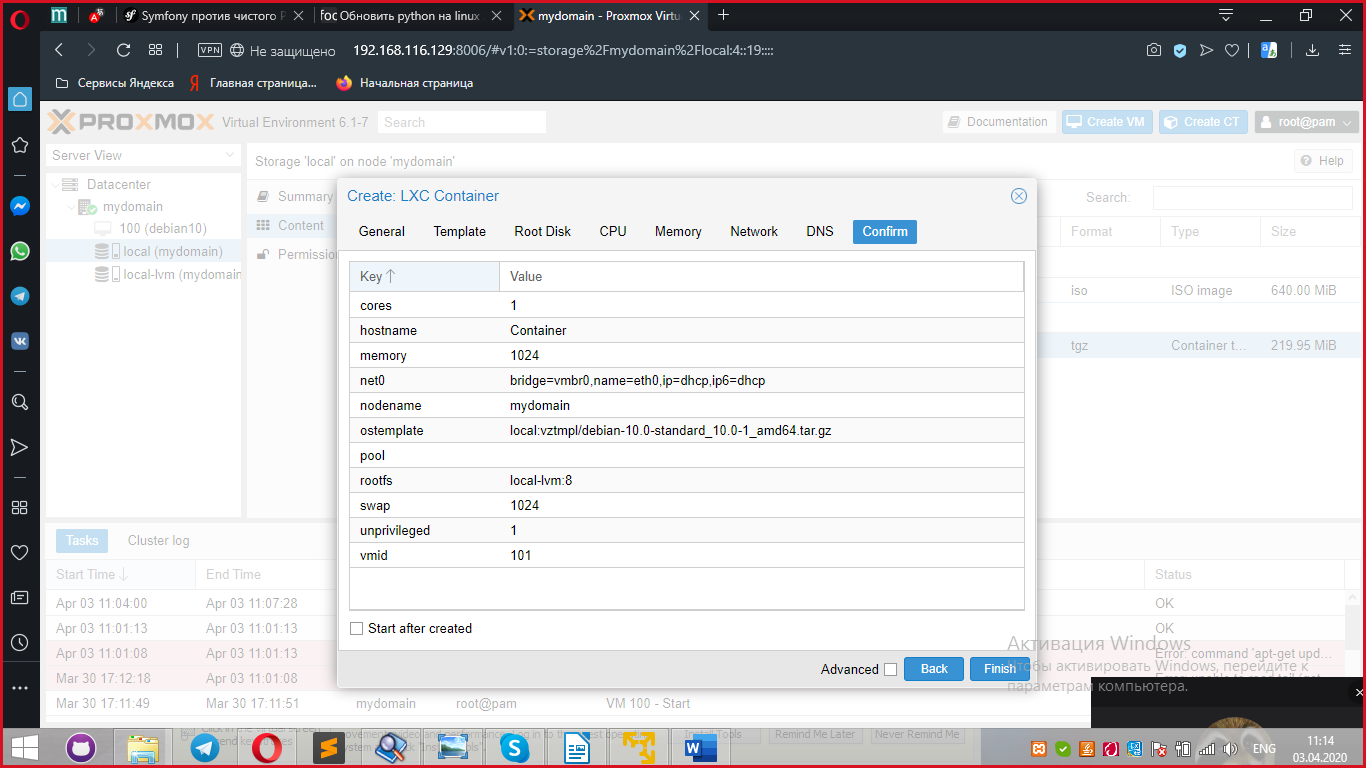
Выделяем ресурсы для контейнера. Размер диска – 8Гб, оперативной памяти – 1Гб, 1 ядро.



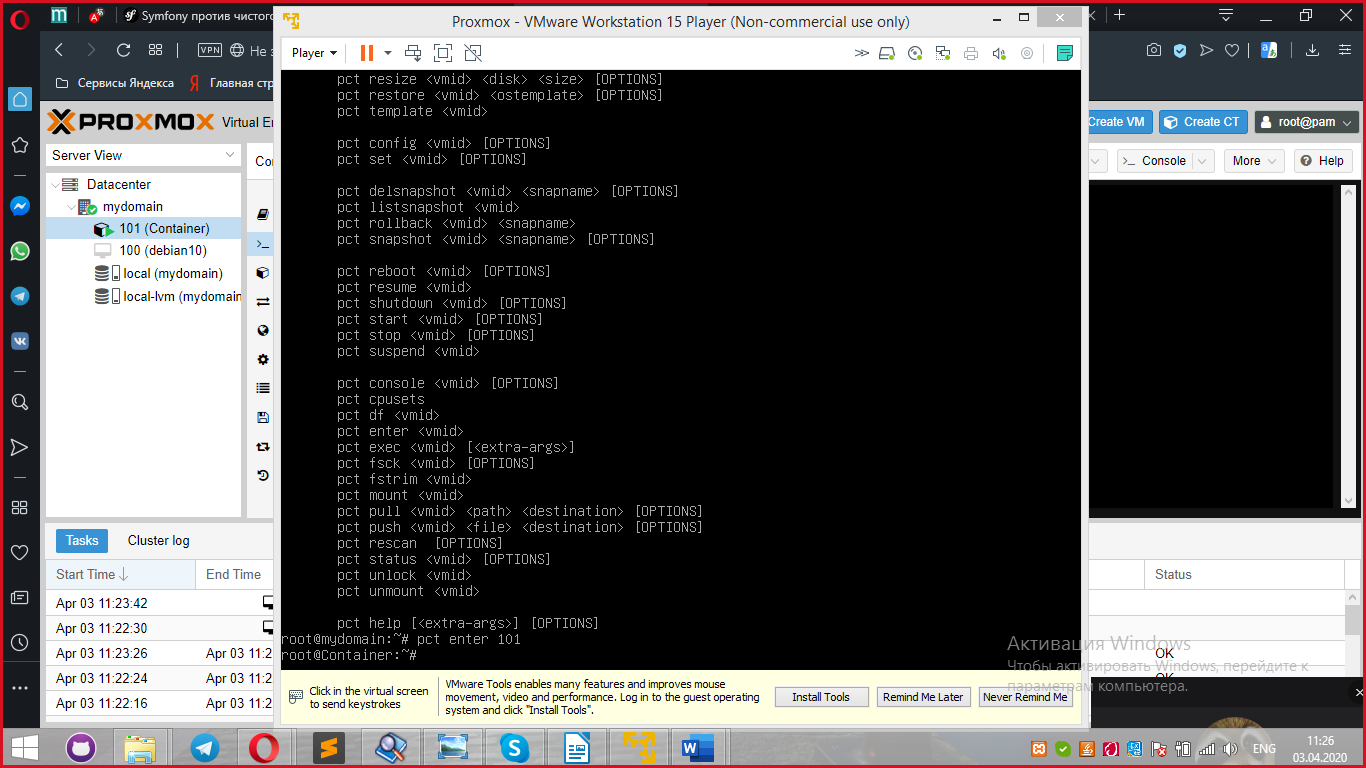


Настроим сеть. Выбираем сетевой мост и протокол DHCP.





Контейнер был успешно создан, запускаем его.

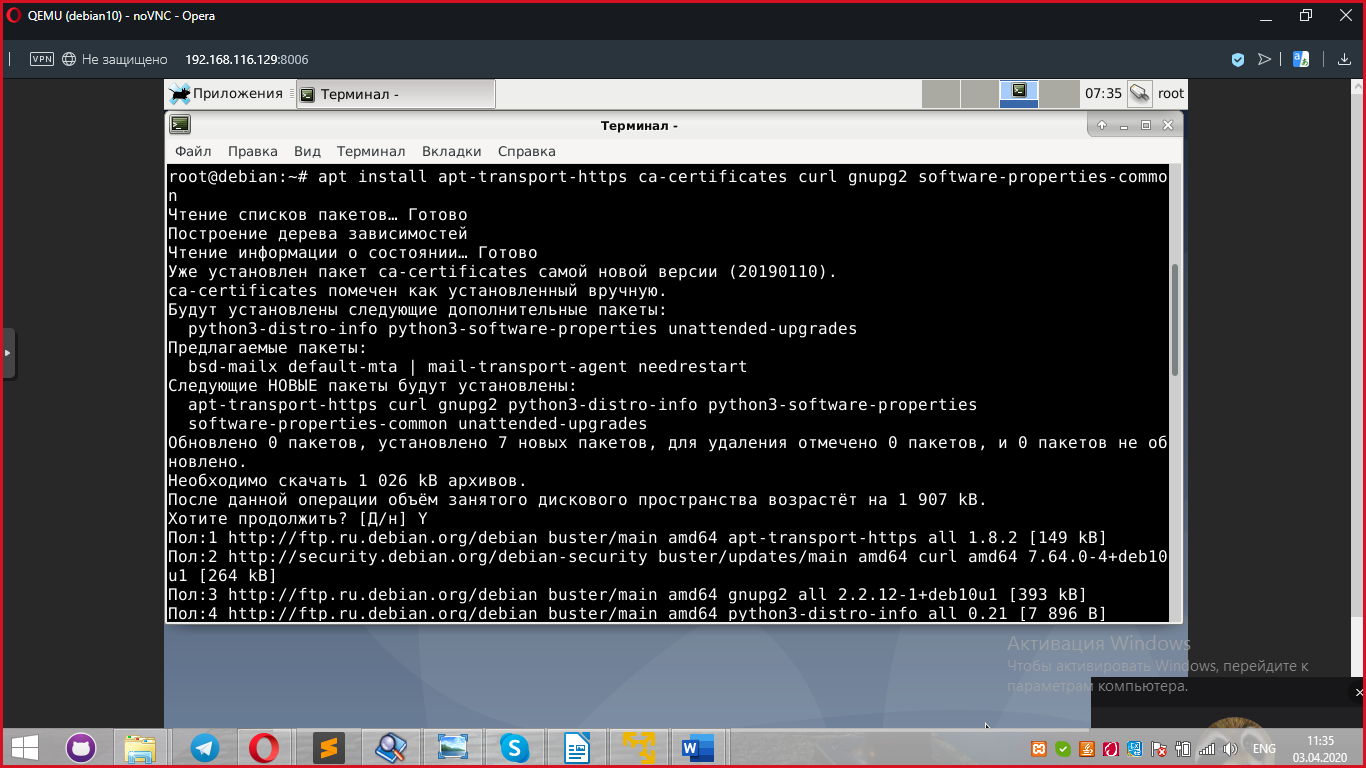


**Docker** — программное обеспечение для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесен на любую Linux-систему с поддержкой cgroups в ядре, а также предоставляет среду по управлению контейнерами.

Изначально использовал возможности LXC, с 2015 года применял собственную библиотеку, абстрагирующую виртуализационные возможности ядра Linux — libcontainer. С появлением Open Container Initiative начался переход от монолитной к модульной архитектуре.

Установим Docker на Debian 10.

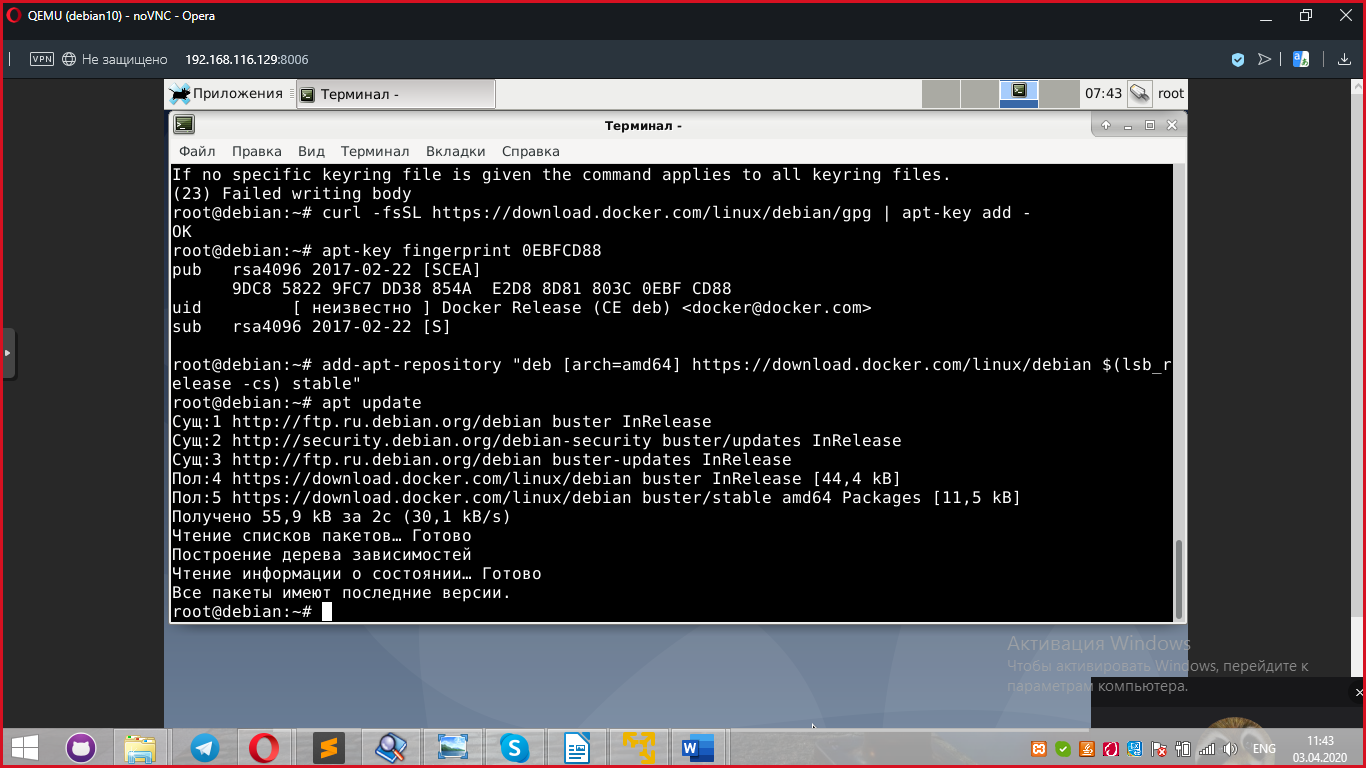
Установим необходимые для работы пакеты командой apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common.



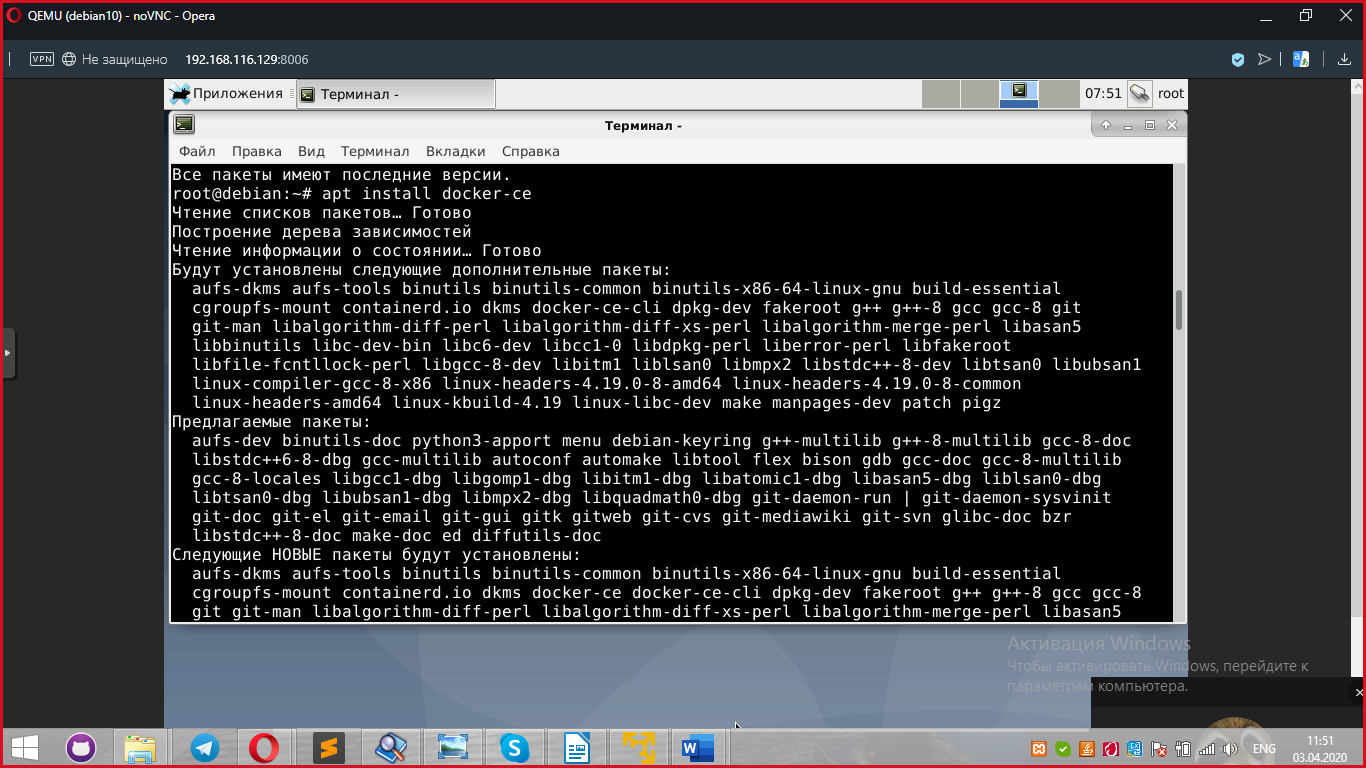
Скачали Docker:



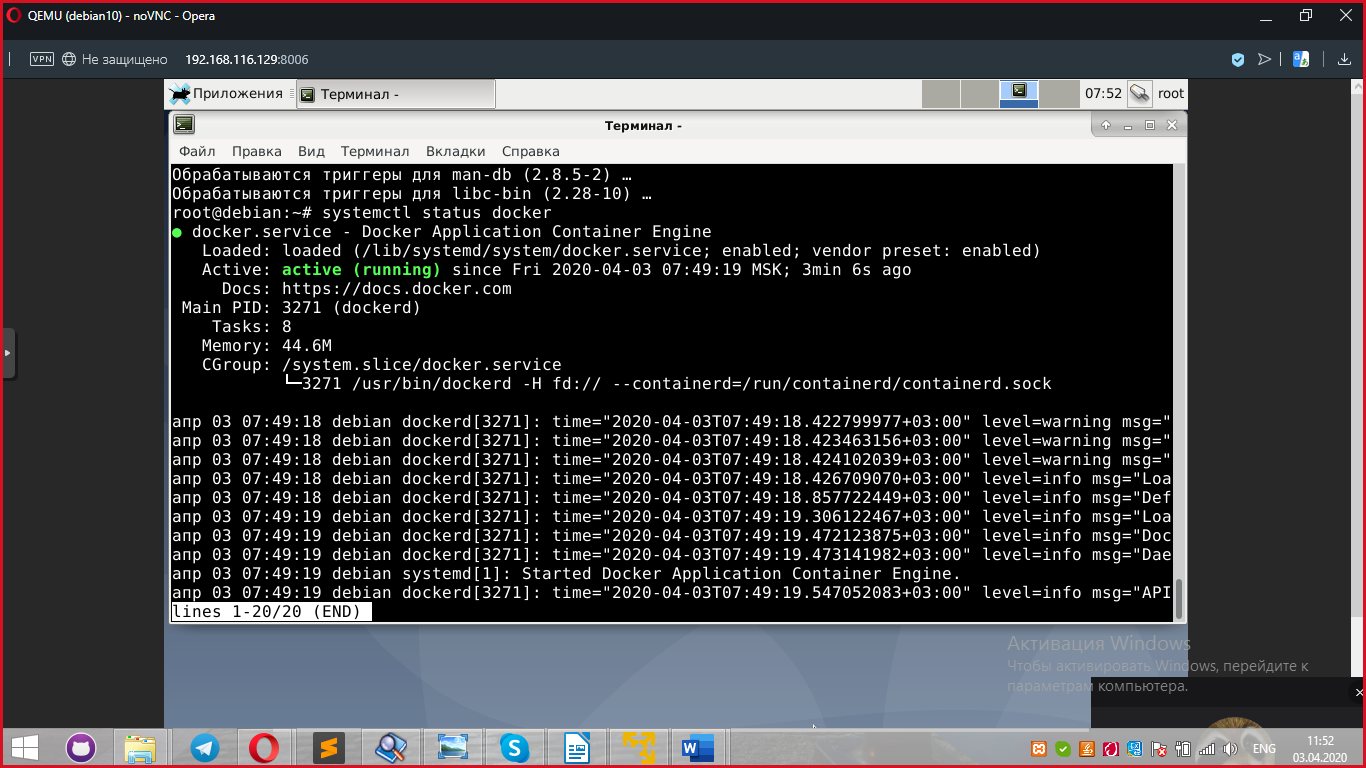
После загрузки добавим данный репозиторий Docker к источникам apt.



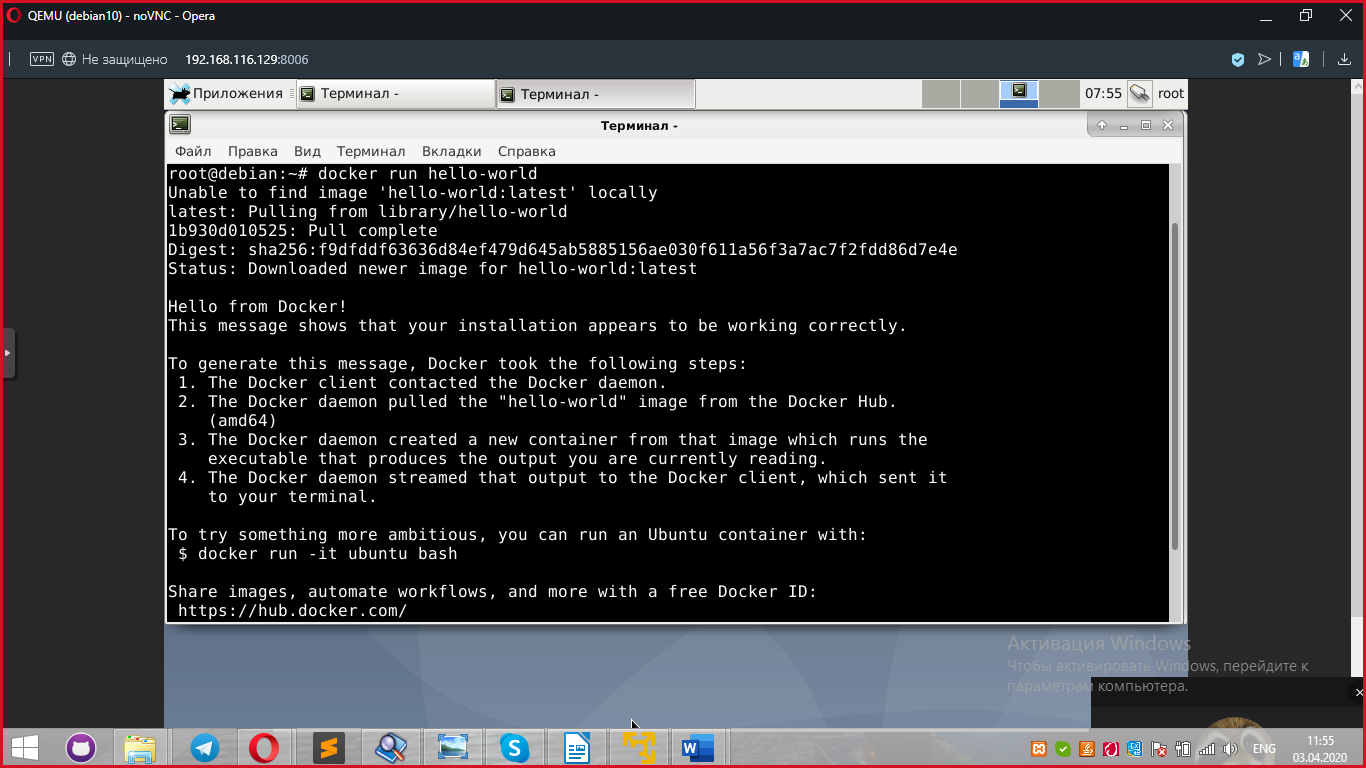
Устанавливаем Docker.



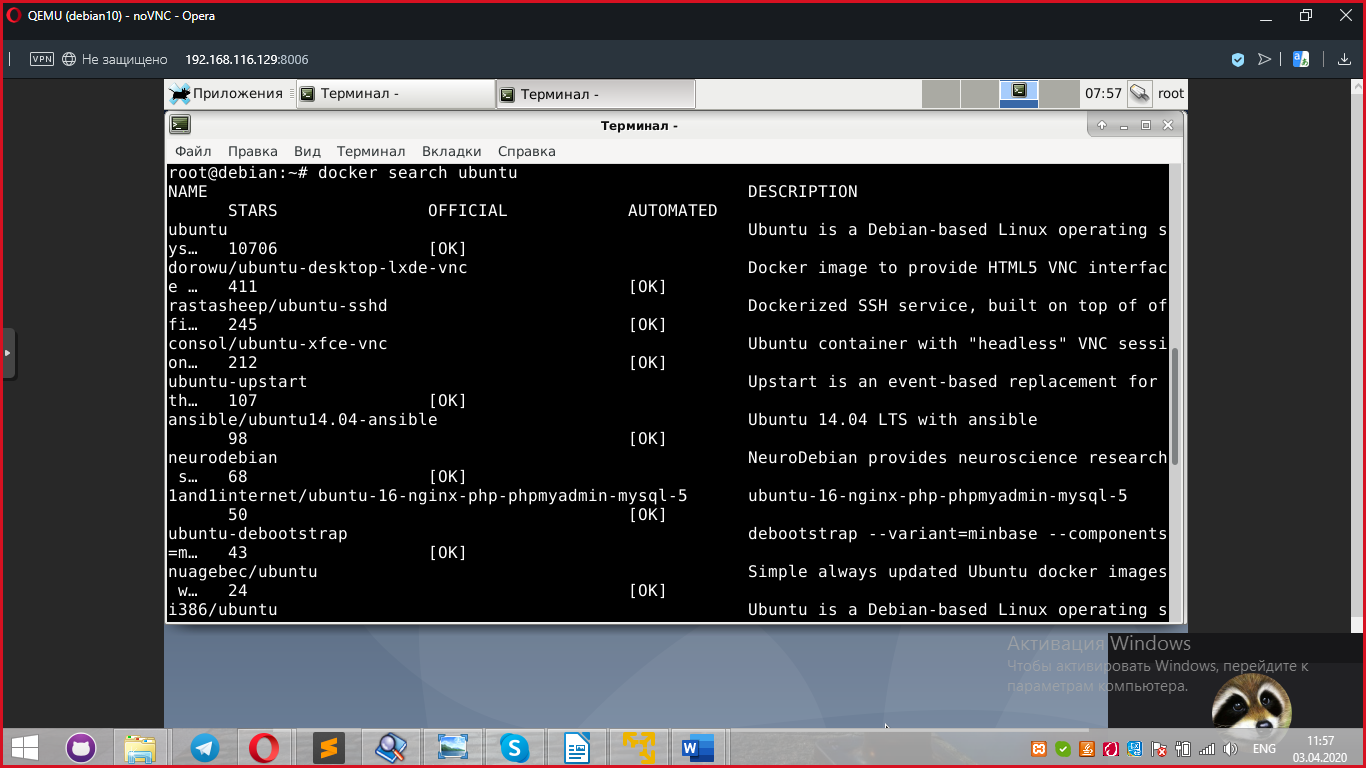
Проверили статус установки с помощью команды systemctl status docker.



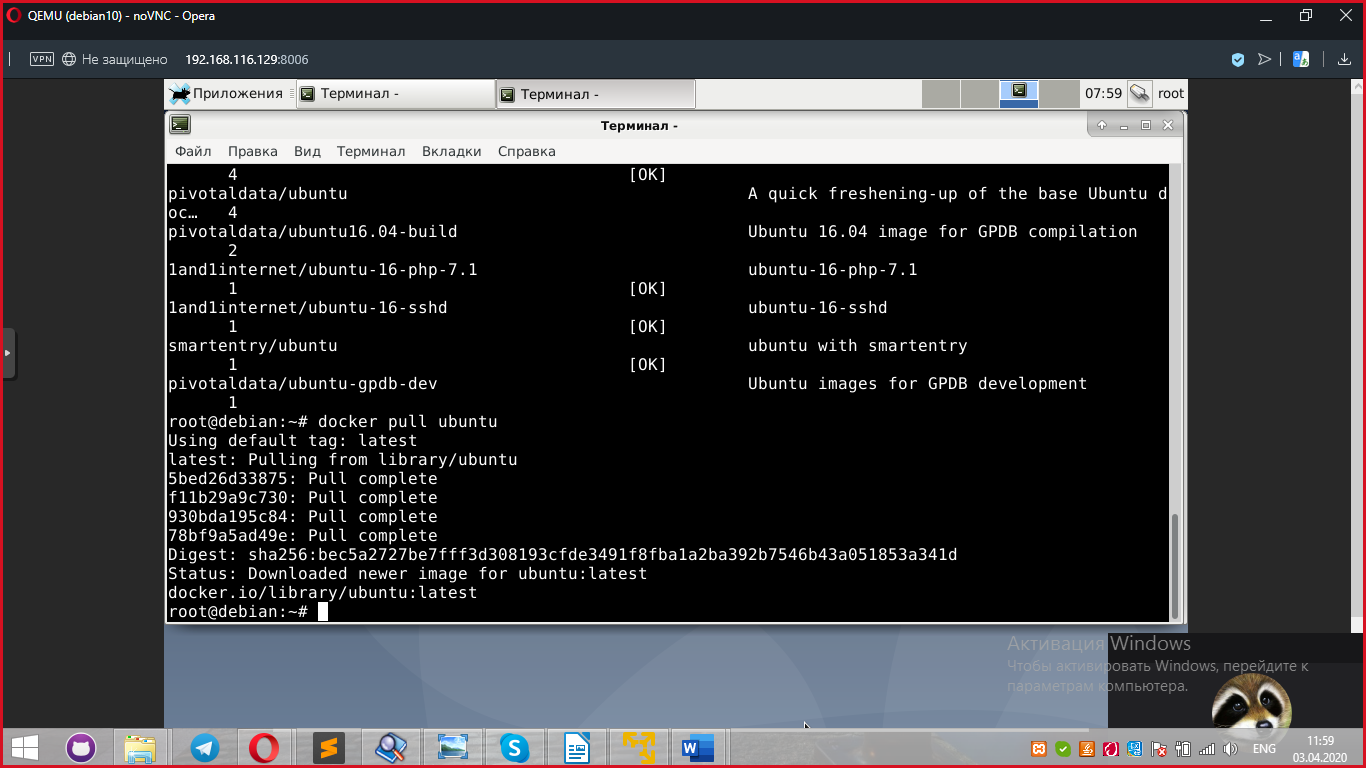
Чтобы проверить, что все работает введем команду *docker run hello-world*.



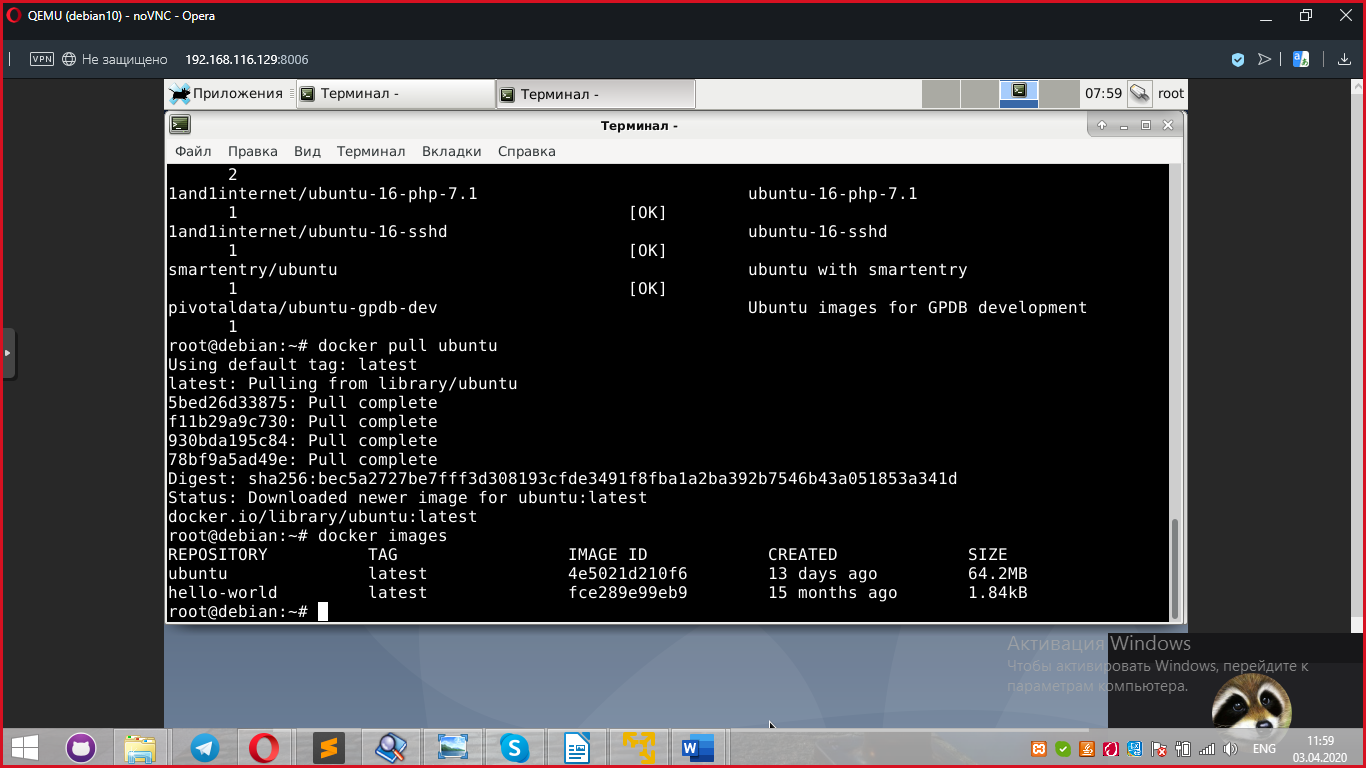
Находим образ Ubuntu с помощью команды *docker search ubuntu.*



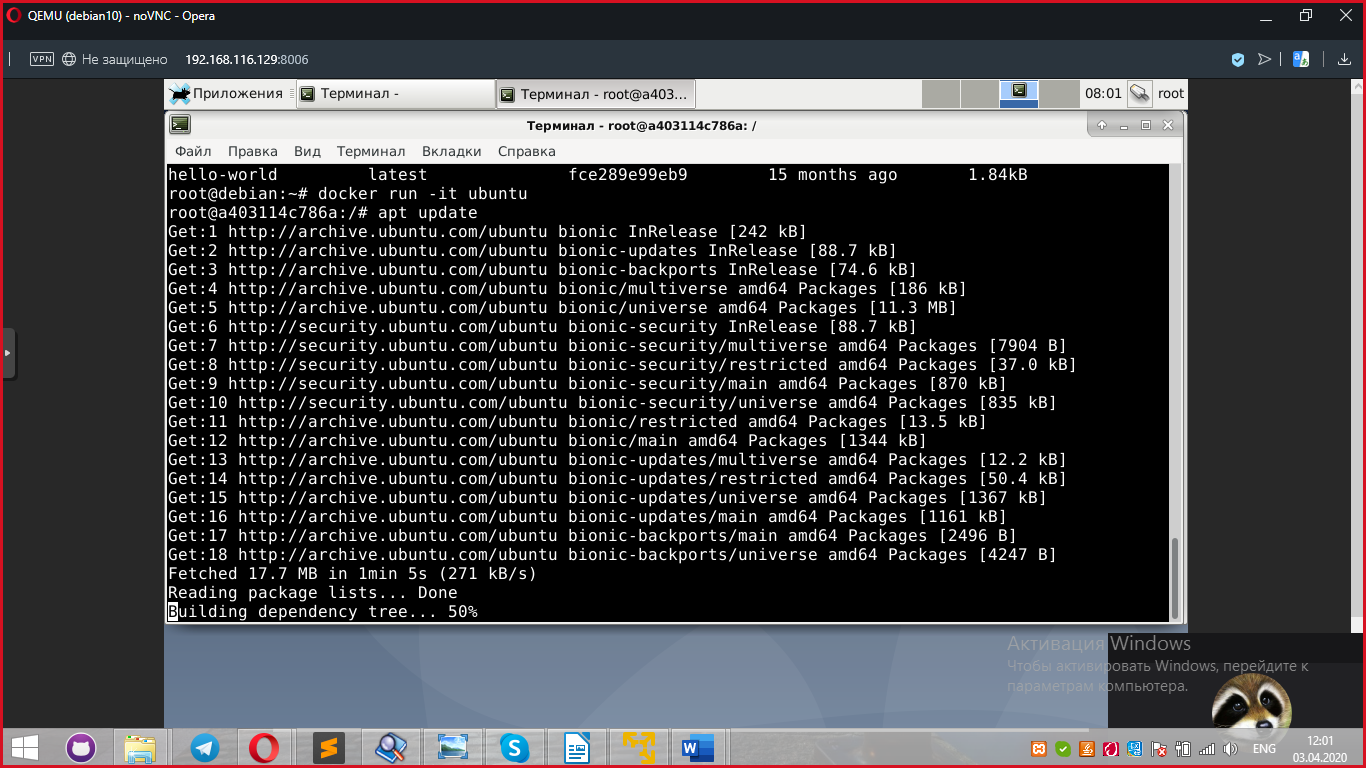
Загружаем образ



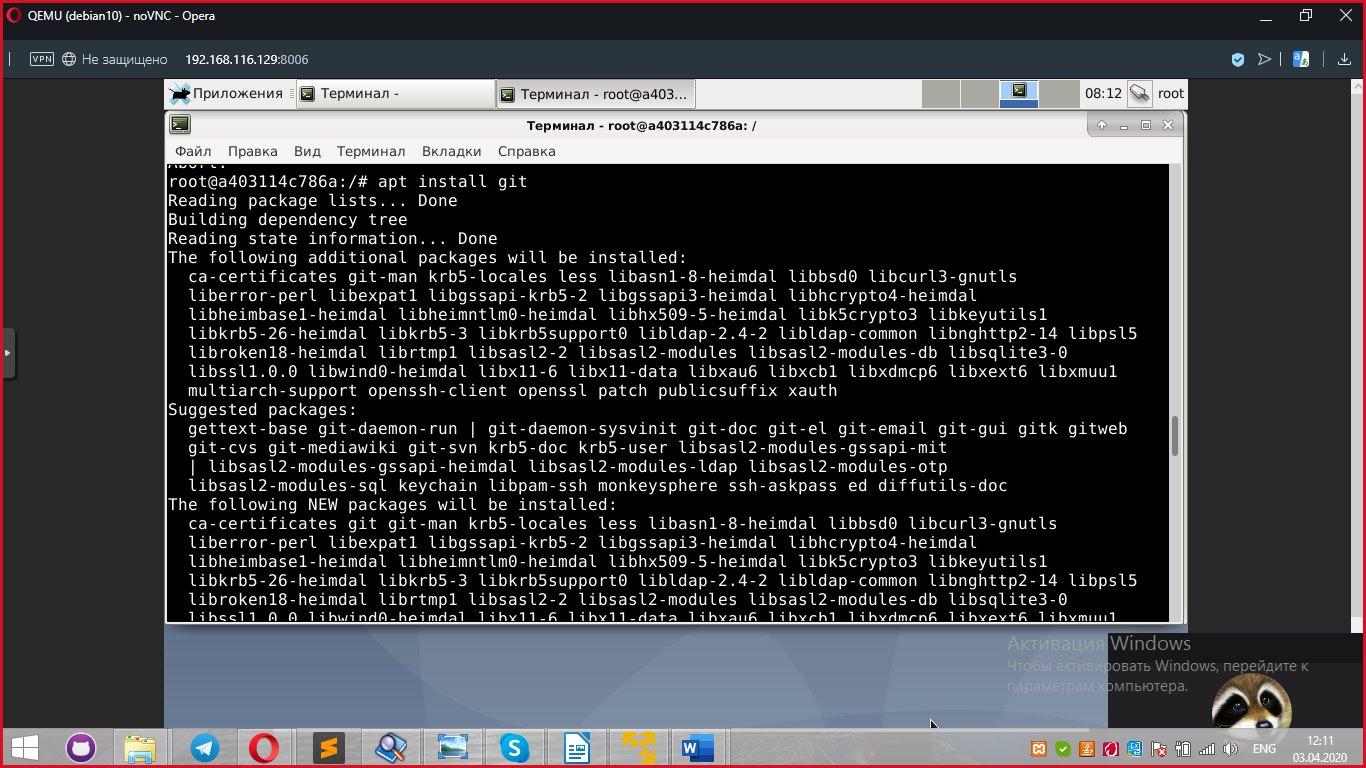
Образ отобразился в списке доступных:

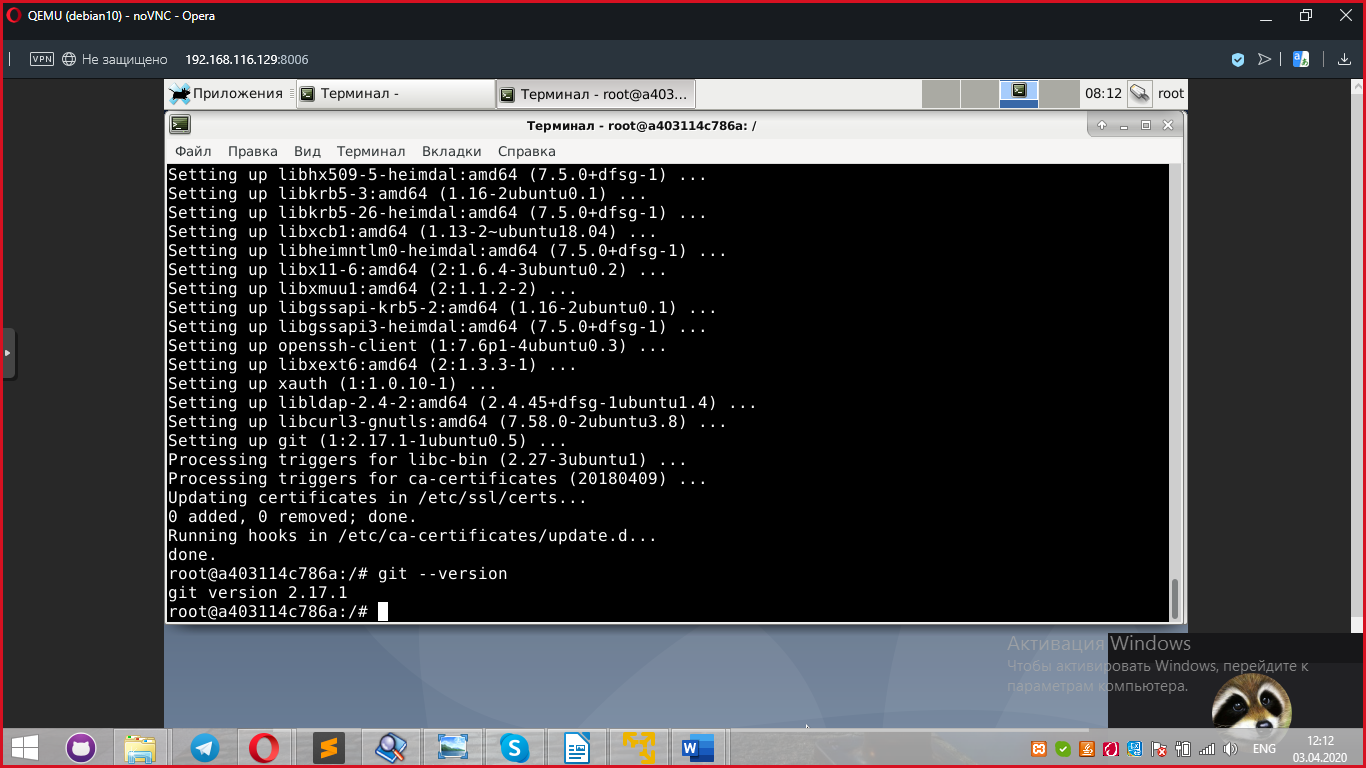


Запустим контейнер:



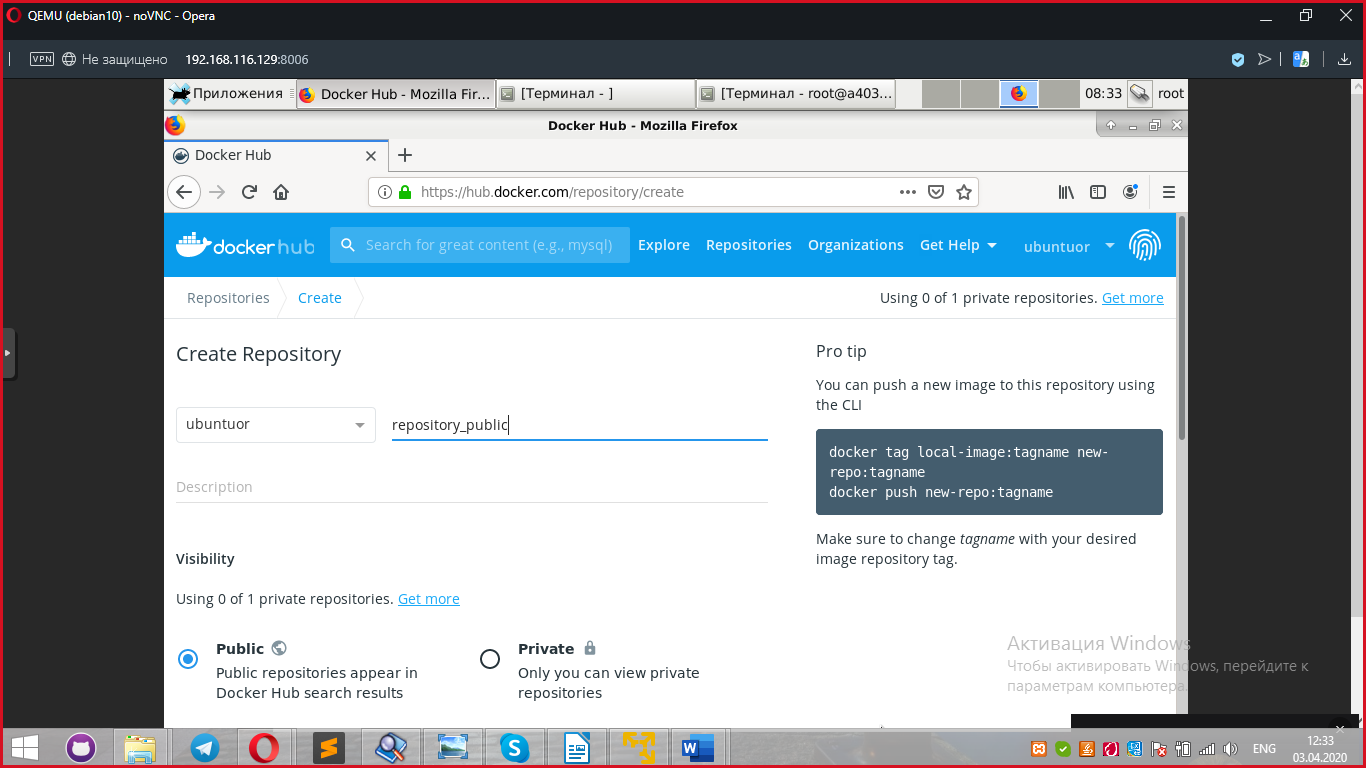
Установили в него git для проверки системы:

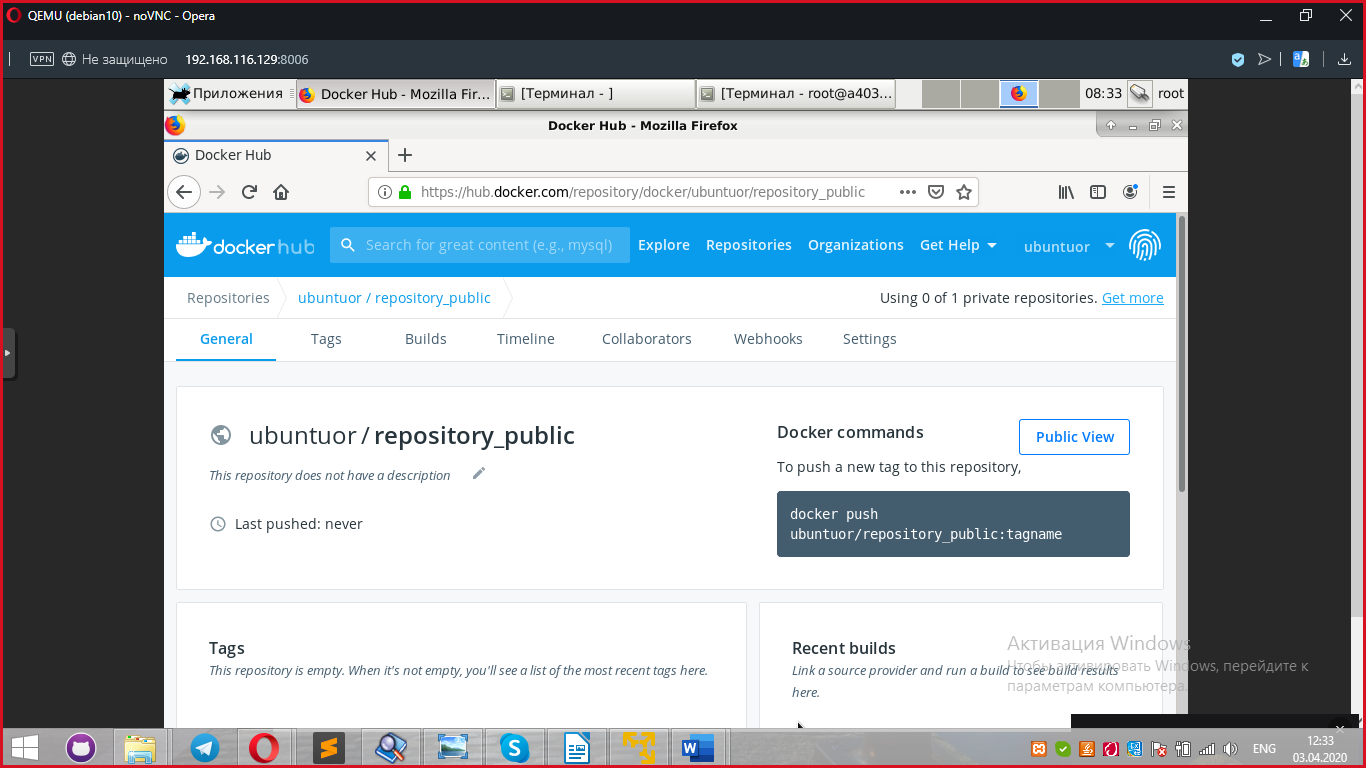




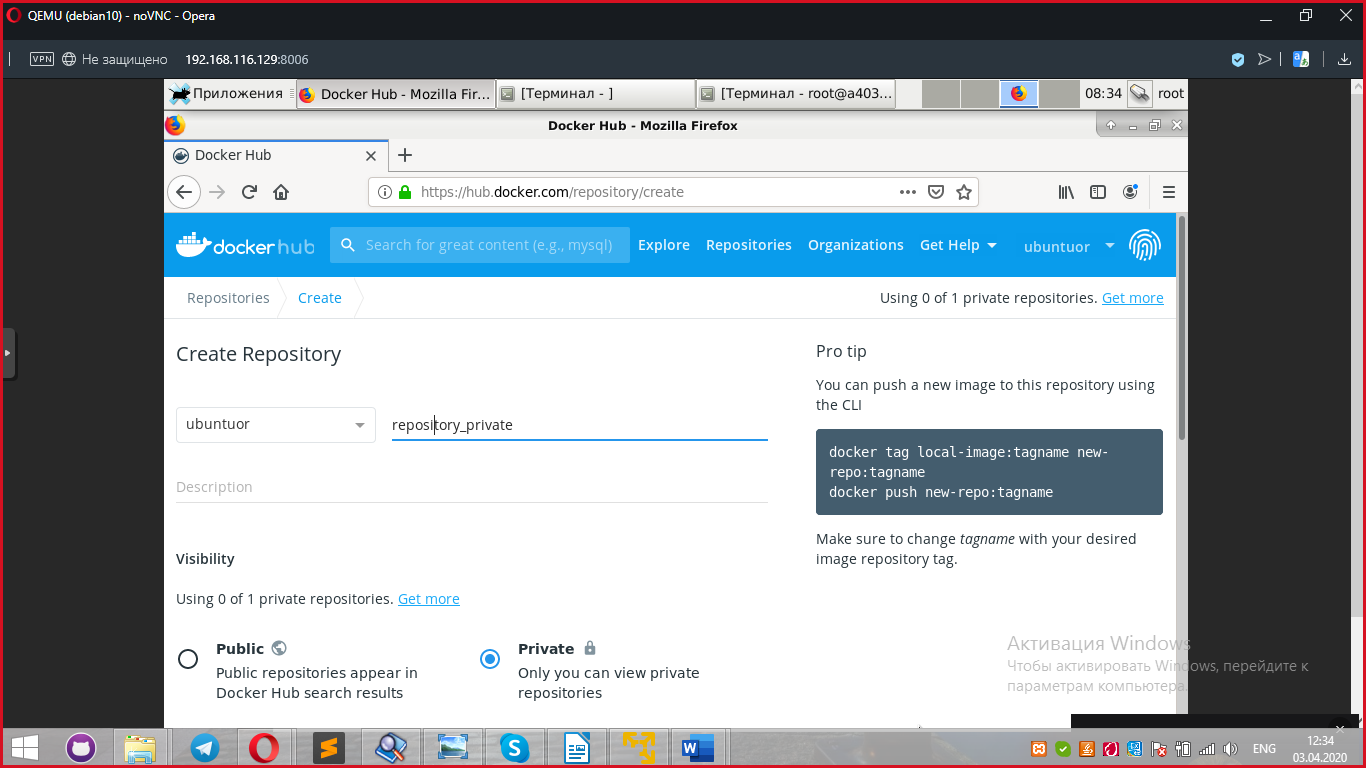
Зарегистрировались и авторизовались в Docer Hub.

Создаем публичный репозиторий:





Создаем приватный репозиторий:





Теперь используя команду push можем загружать образы в репозитории. Если захотим выгрузить на другой linux машине образ, то репозиторий нужно сделать открытым и использовать команду pull.

