МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное   
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики  
Кафедра суперкомпьютеров и общей информатики

**Отчет по лабораторной работе №2**

Дисциплина: «Развертывание и жизненный цикл программного обеспечения»

Тема: **«Git and CI»**

Выполнил: Мелешенко И.С.

Группа: 6133-010402D

Самара 2021

**TASK**

Шаги

1. Download Gitlab-Bitnami vm image from https://bitnami.com/stack/gitlab/virtual-machine
2. Upload https://github.com/olindata/sample-gitlabci-cpp-project to your Gitlab server.
3. To unblock SSH https://docs.bitnami.com/virtual-machine/faq/get-started/enable-ssh/
4. https://askubuntu.com/questions/204400/ssh-public-key-no-supported-authentication-methods-available-server-sent-publ
5. Install GitLab Runner using the official GitLab repositories https://docs.gitlab.com/runner/install/linux-repository.html
6. Update /etc/gitlab/gitlab.rb to disable https on gitlab (yes, it is not for production)

# use here your IP, but is must be HTTP

external\_url 'http://192.168.88.228'

nginx['redirect\_http\_to\_https'] = false

nginx['ssl\_verify\_client'] = "off"

1. Reconfigure GitLab for the changes to take effect:

$ sudo gitlab-ctl reconfigure

1. Register runner. Choose shell executor type. Use your ip and registration-token for command below:

$ sudo gitlab-runner register --url http://192.168.88.228/ --registration-token yqjsLYNFrbjaC-QhmycE

1. Edit .gitlab-ci.yml to run runner in shell mode (without Docker)

job:

script:

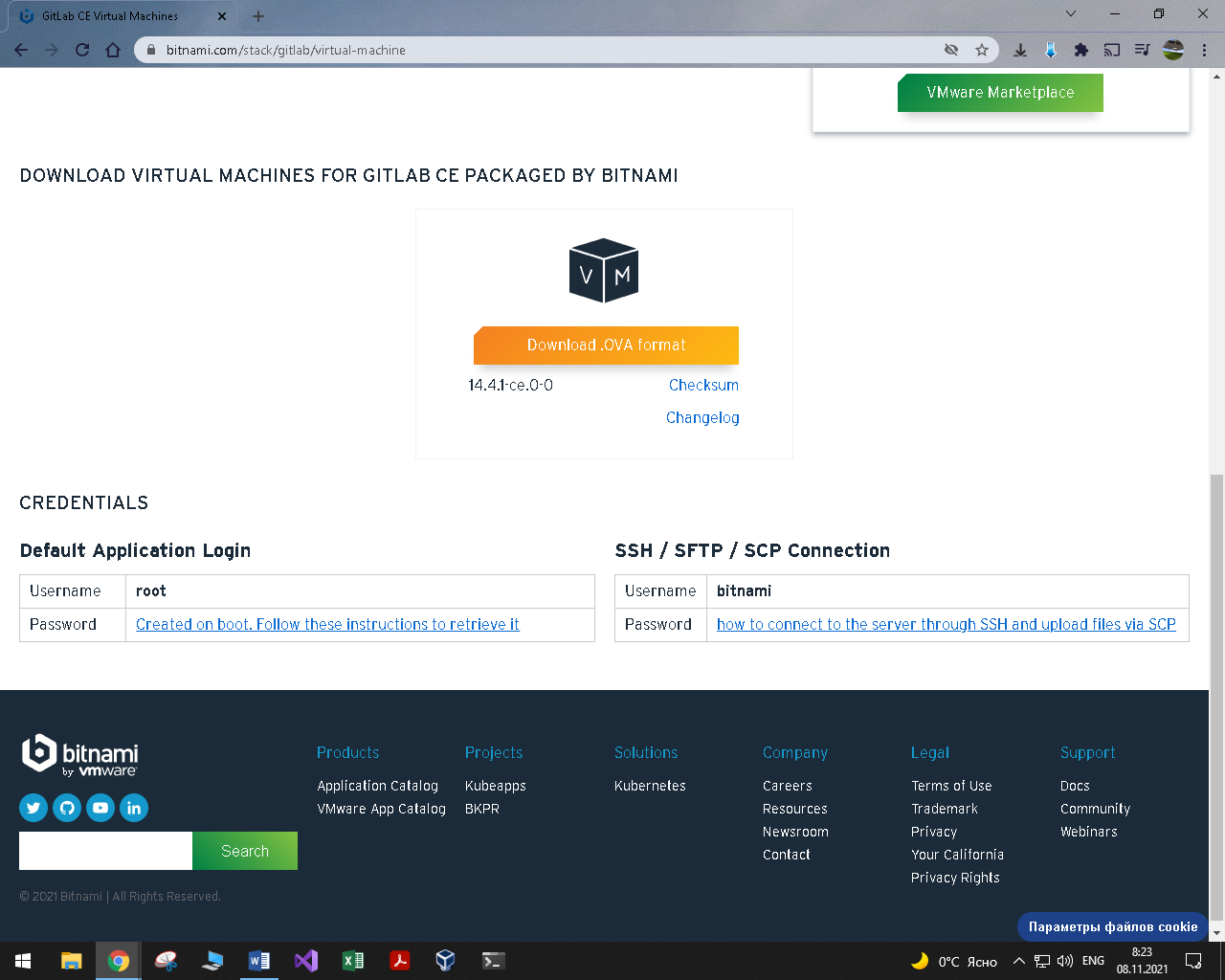
g++ helloworld.cpp -o helloworld

1. Run Pipeline: CI/CD > Pipelines > Run pipeline

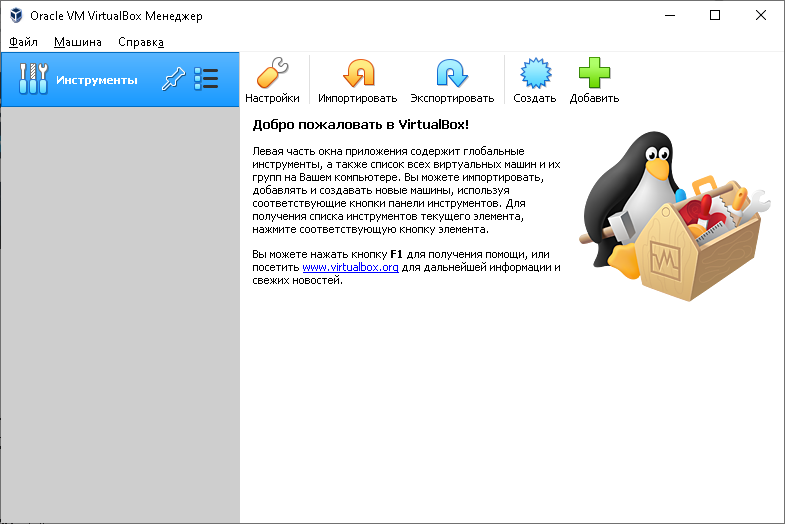
ХОД РАБОТЫ

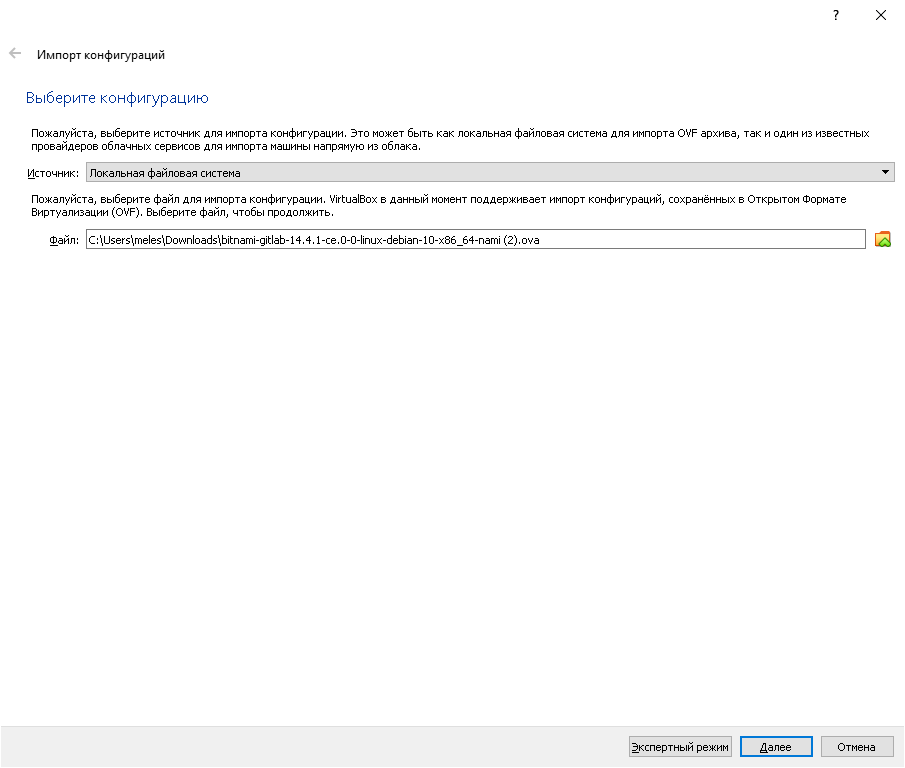
Шаг 1. Загрузка и установка образа виртуальной машины Gitlab-Bitnami

1.1. Загрузим образ виртуальной машины с сайта https://bitnami.com/stack/gitlab/virtual-machine.

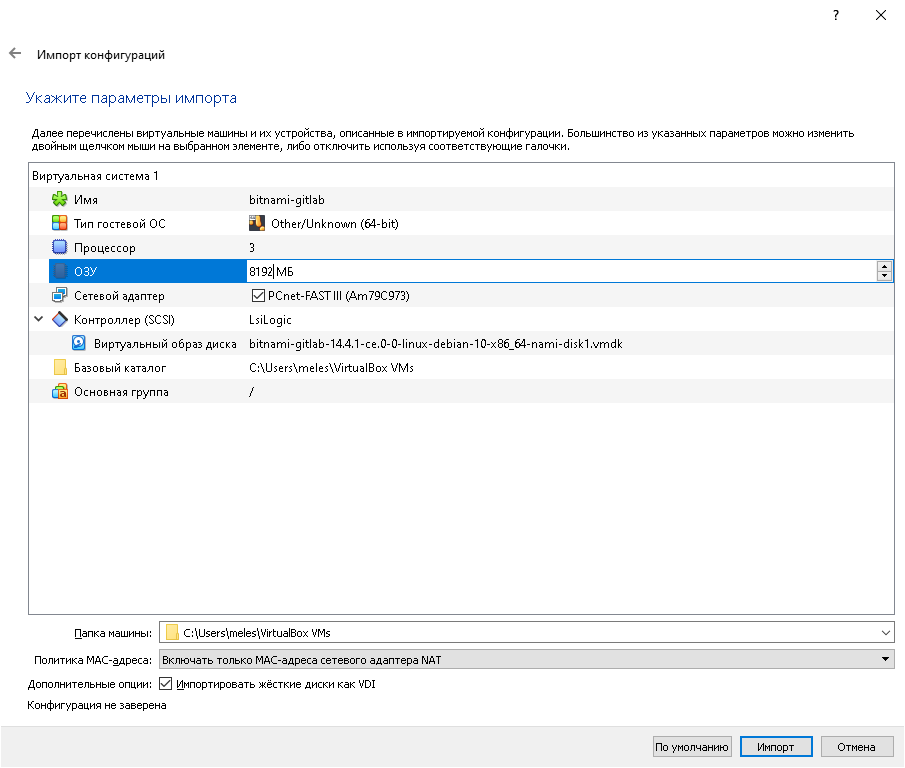


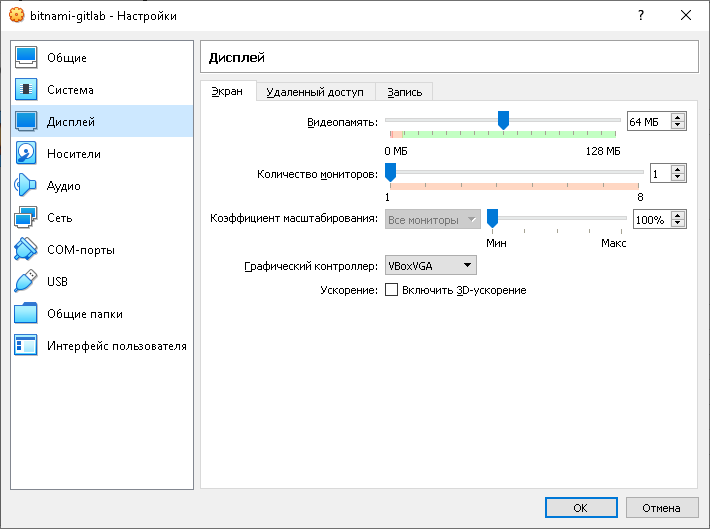
1.2. Импортируем скачанный образ в VirtualBox.



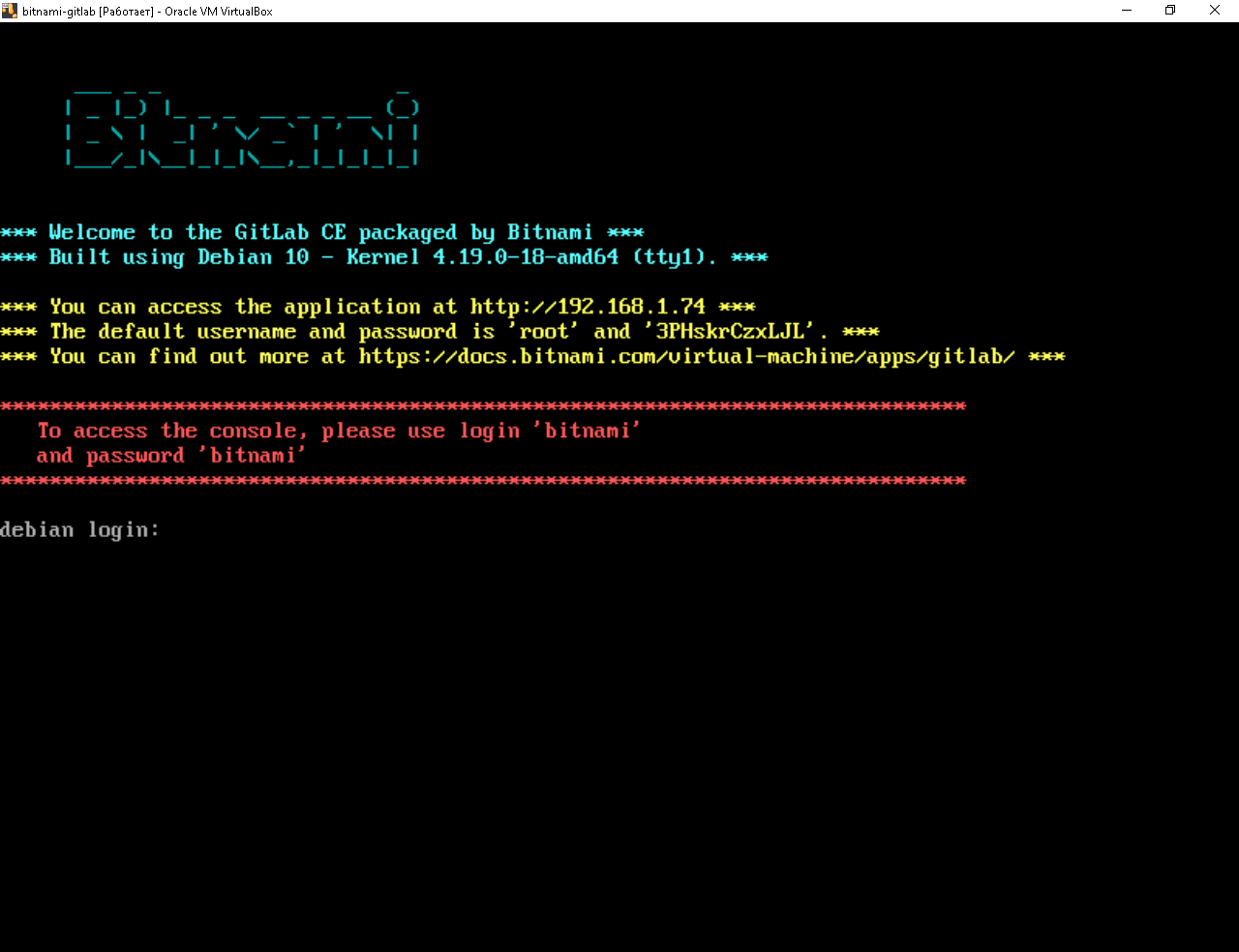


1.3. Настроим виртуальную машину. В качестве процессора выделим ей ровно половину ресурсов ПК, то есть 3 физических ядра процессора и половину объема оперативной памяти – 8192 Мб, а также выделим ей 64 Мб видеопамяти.





1.4 Запустим настроенную виртуальную машину.



На данном этапе будет необходимо скопировать несколько данных, которые нам в будущем очень пригодятся:

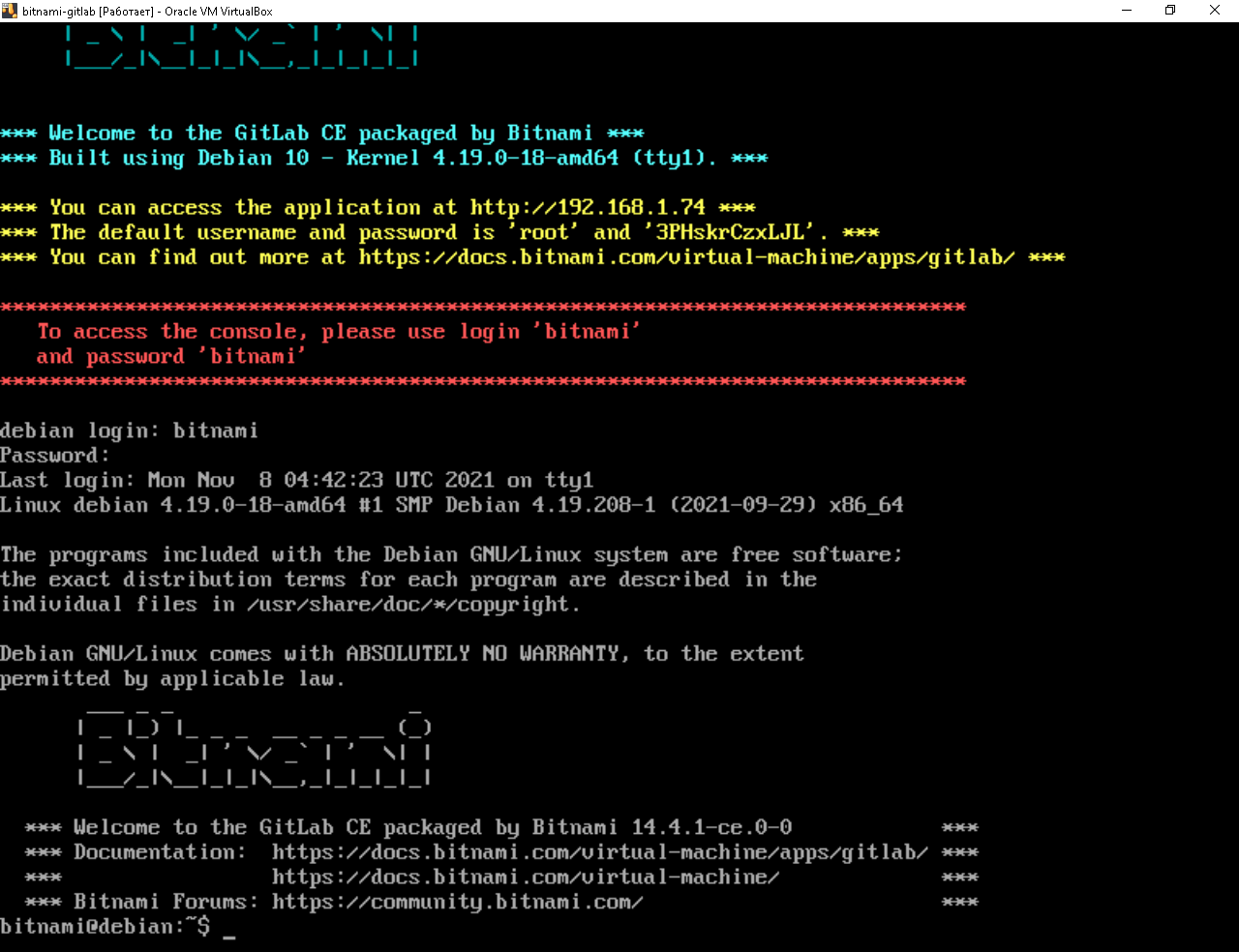
Адрес приложения - <https://192.168.1.74>;

Username – root;

Password – 3PHskrCzxLJL;

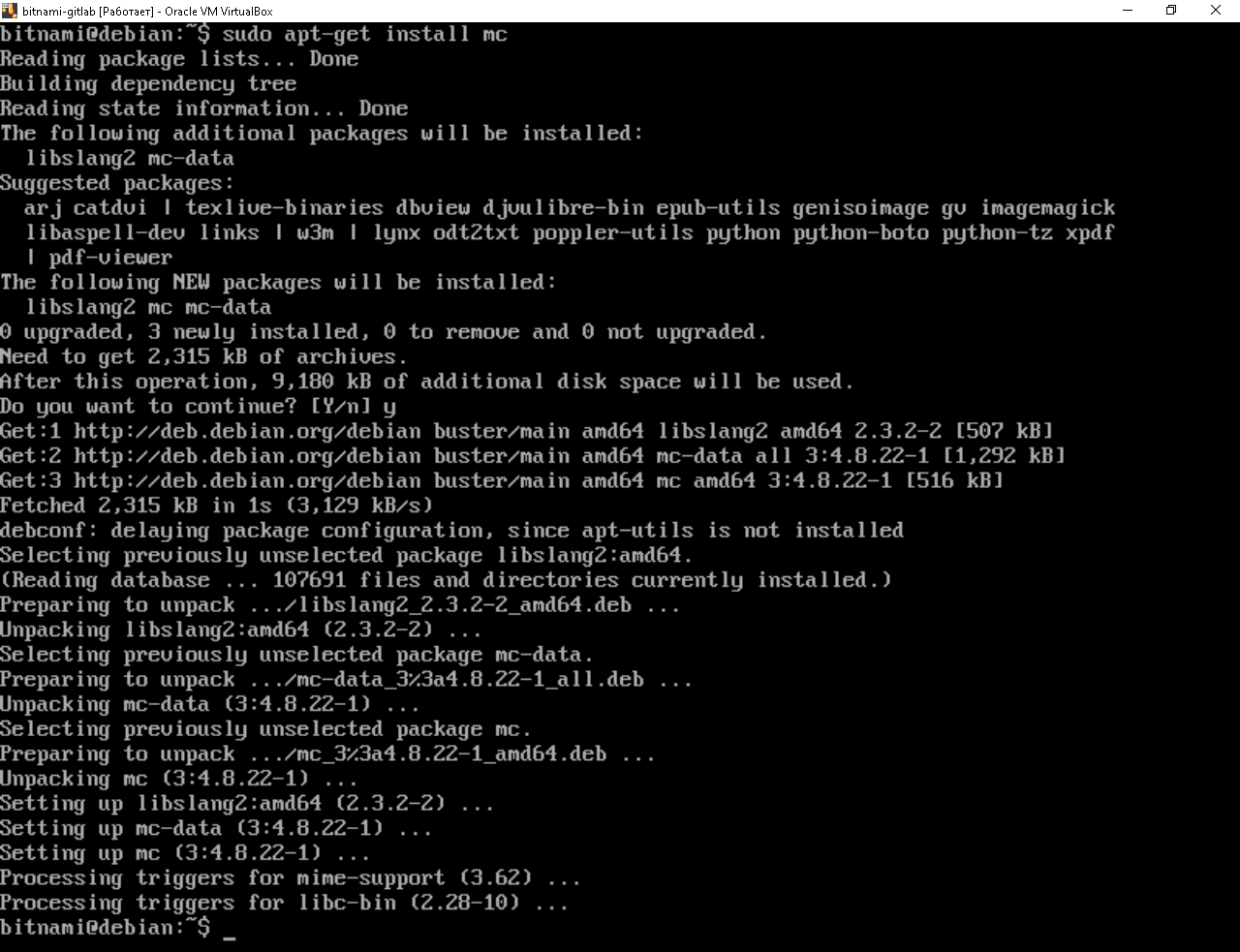
1.5 Выполним вход на сервер приложения.

По умолчанию логин и пароль для входа на сервер являются bitnami и bitnami.

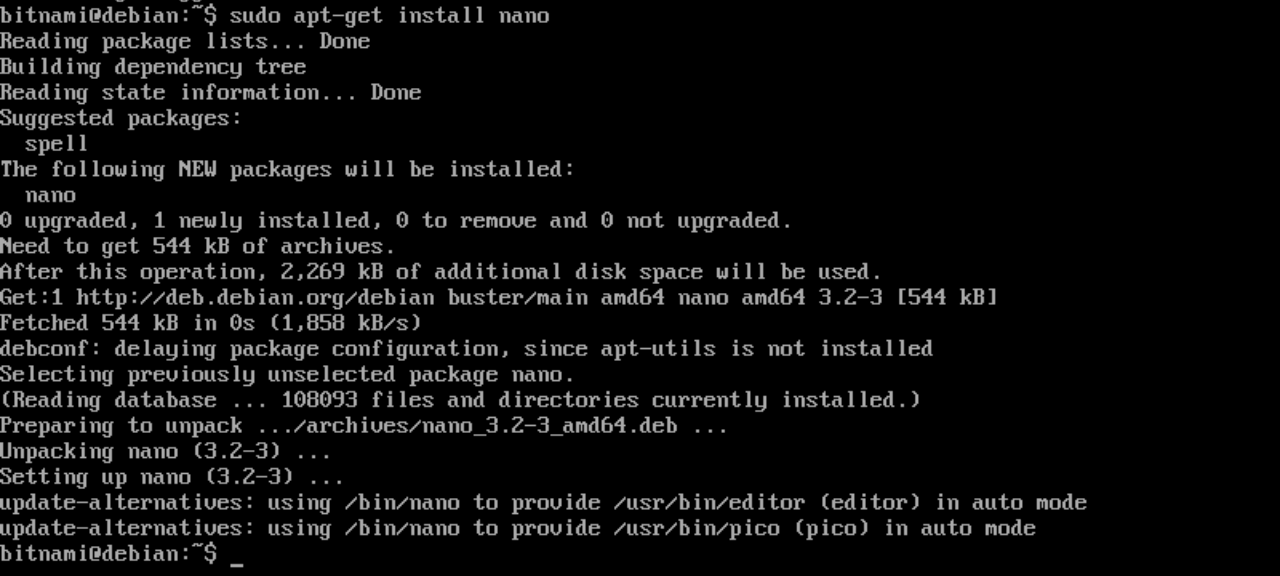


После выполнения входа, необходимо будет подождать продолжительное время для развертывания и инициализации всей машины. А в это время произведем установку пары элементов, которые пригодятся позже.

1. sudo apt-get install mc – установка аналога Total Commander.

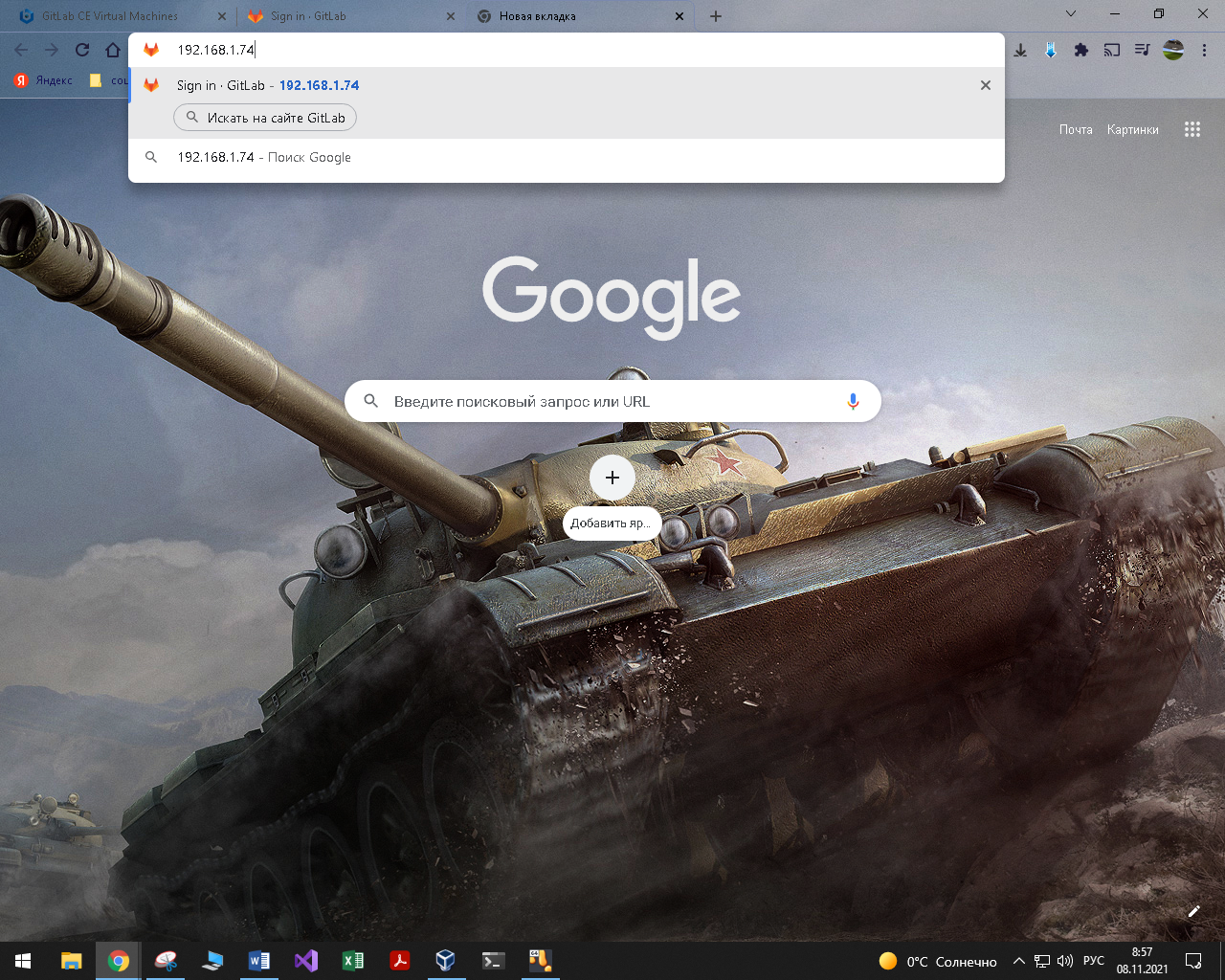


1. sudo apt-get install nano – установка аналога текстового редактора.

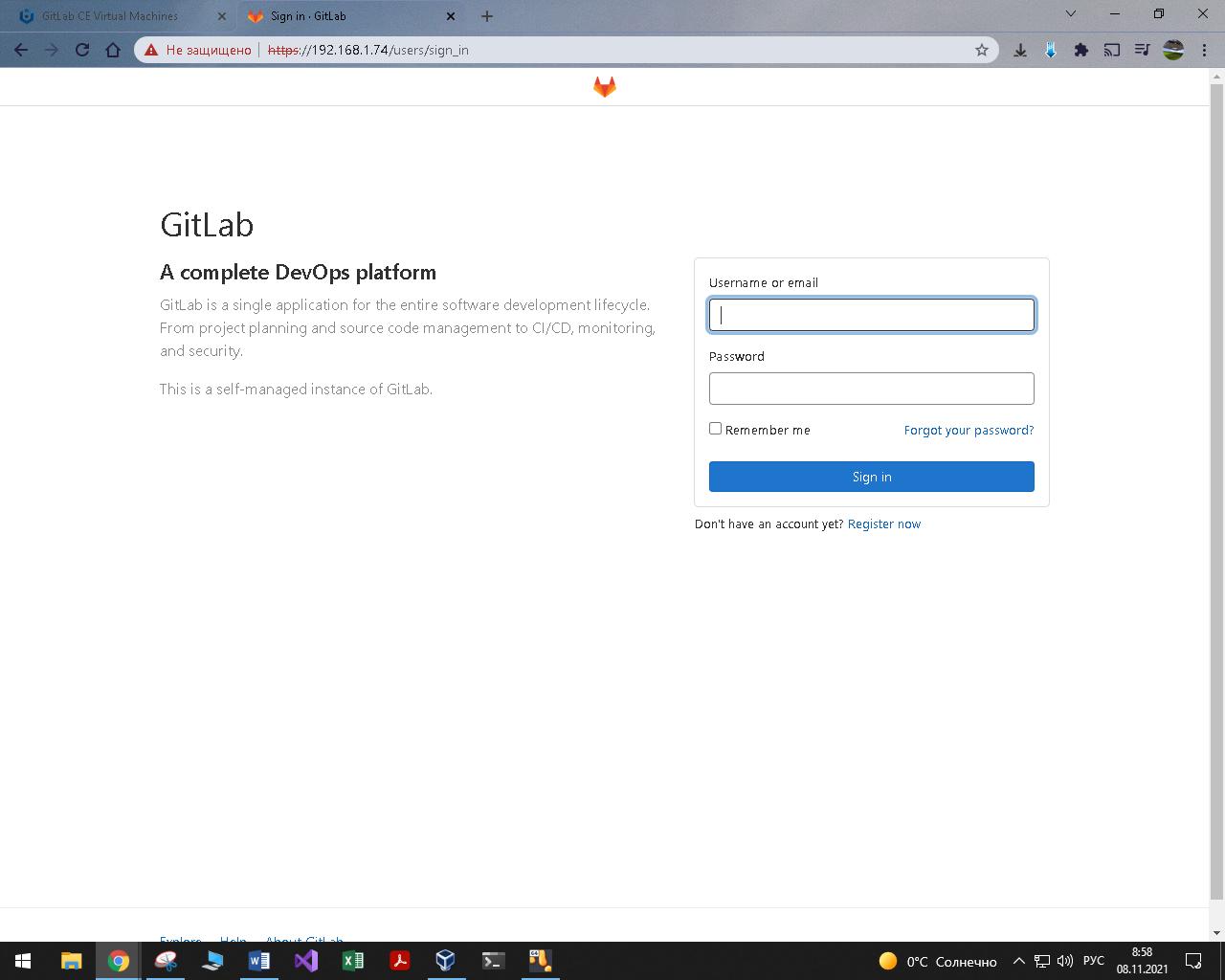


1.6 Выполним вход в приложение.

Перейдем в браузер и в поисковой строке наберем адрес приложения, скопированный с виртуальной машины на шаге 1, при выполнении действия (1.4)



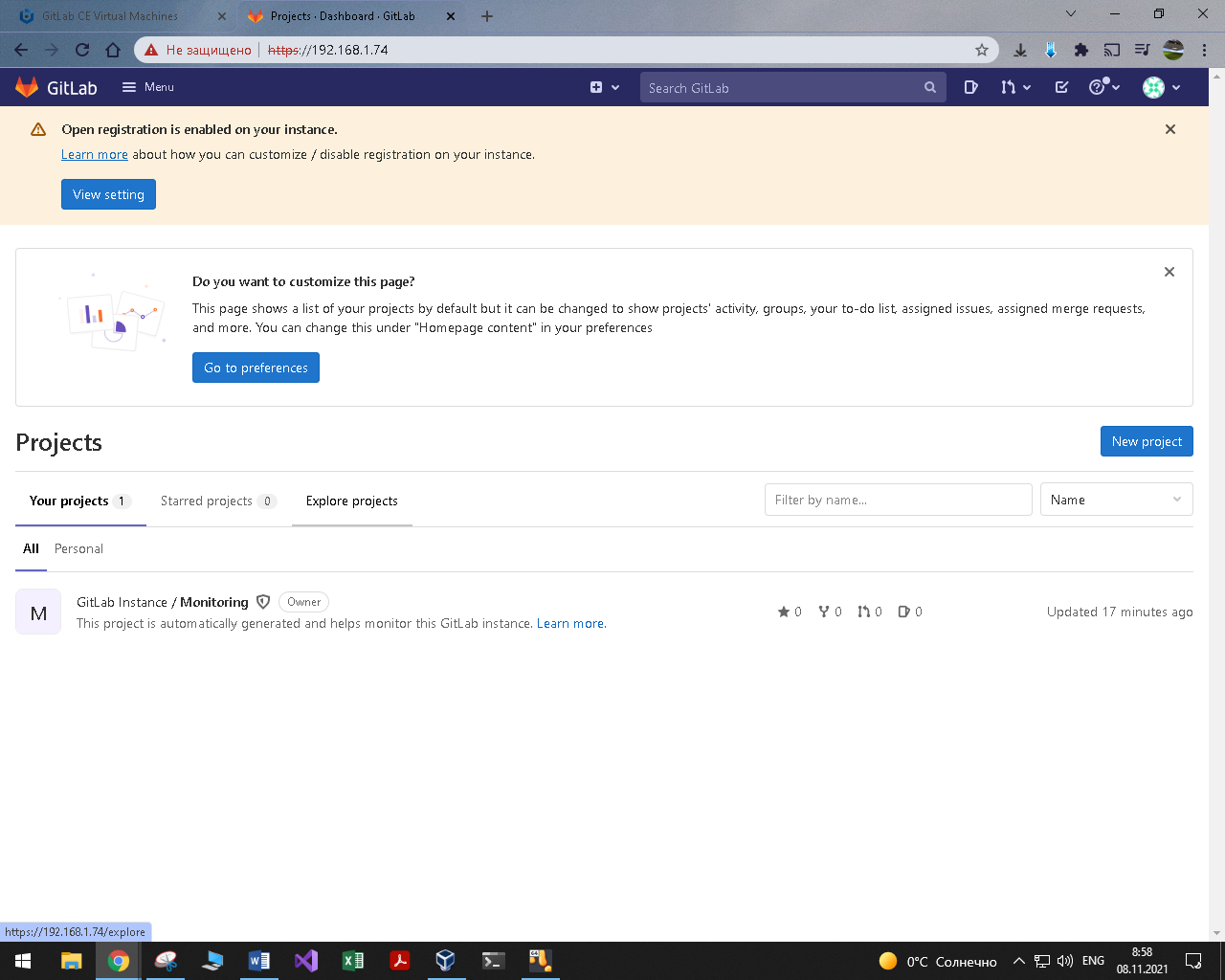
После перехода, нас выведет на страницу идентификации.



Тут необходимо указать логин и пароль, скопированные из виртуальной машины на шаге 1, при выполнении действия 1.4.

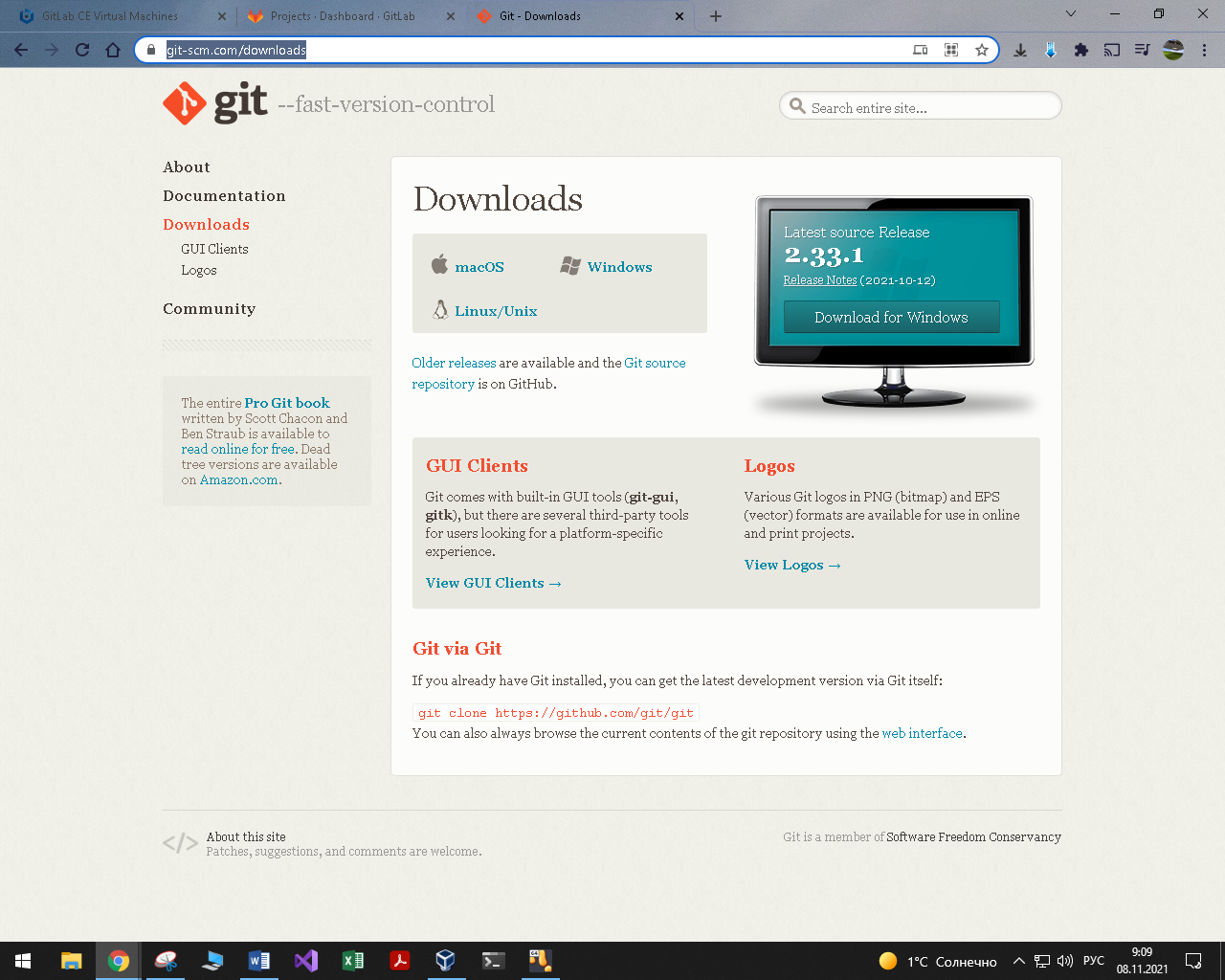


После чего нас выведет на страницу приложения Gitlab, в которой мы и будем работать.

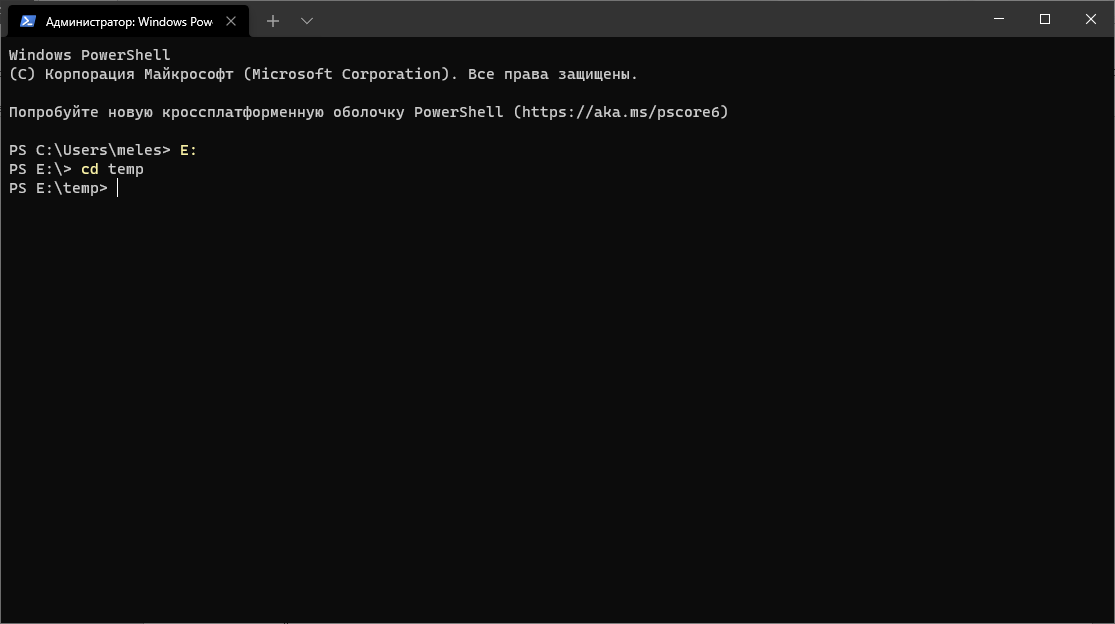


Шаг 2. Загрузка проекта с https://github.com/olindata/sample-gitlabci-cpp-project на наш сервер Gitlab.

2.1 Установка git-scm. Git-scm нам потребуется для продолжения работы, установка данной программы позволит работать с GitHub и Gitlab в консоли (Windows Terminal). Скачать git-scm можно по ссылке https://git-scm.com/downloads

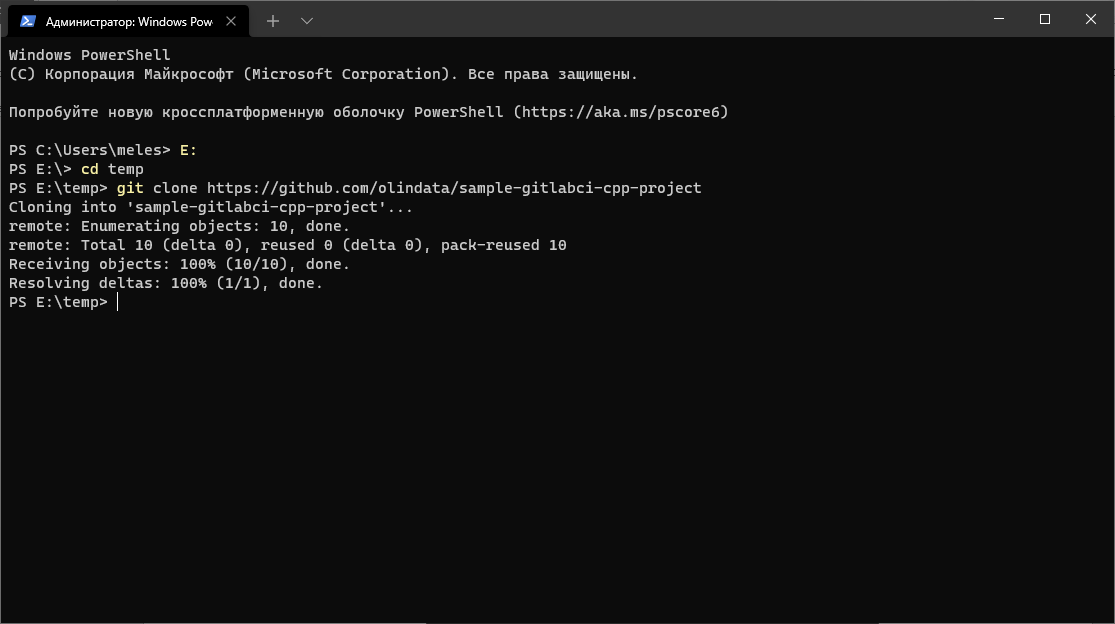


2.2 После установки git-scm, откроем терминал и перейдем в пустую папку в которую будем клонировать репозиторий с сайта https://github.com/olindata/sample-gitlabci-cpp-project

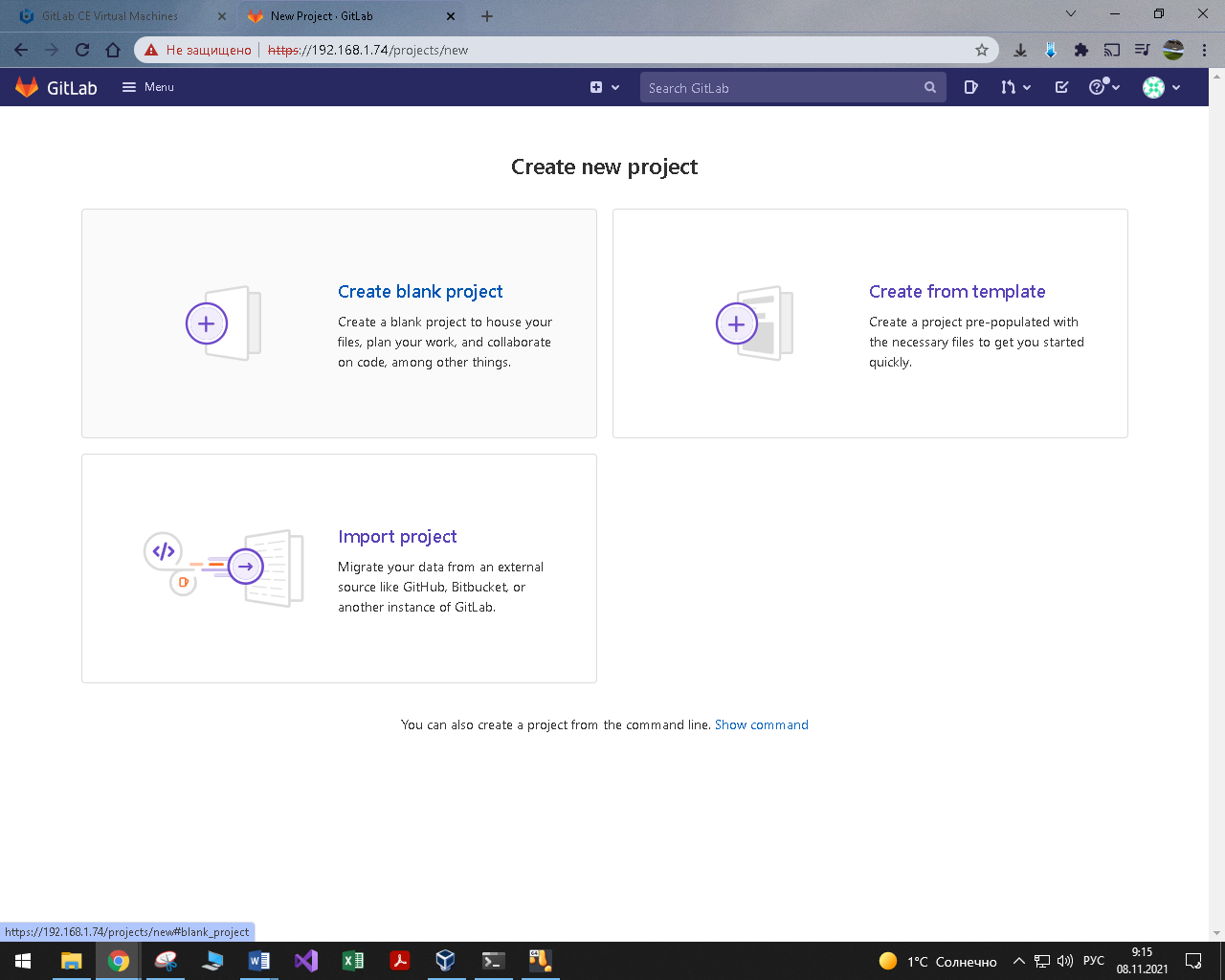


2.3 Клонируем репозиторий в эту папку, при помощи следующей команды:

git clone https://github.com/olindata/sample-gitlabci-cpp-project



2.4 Создадим пустой репозиторий в приложении на сервере.

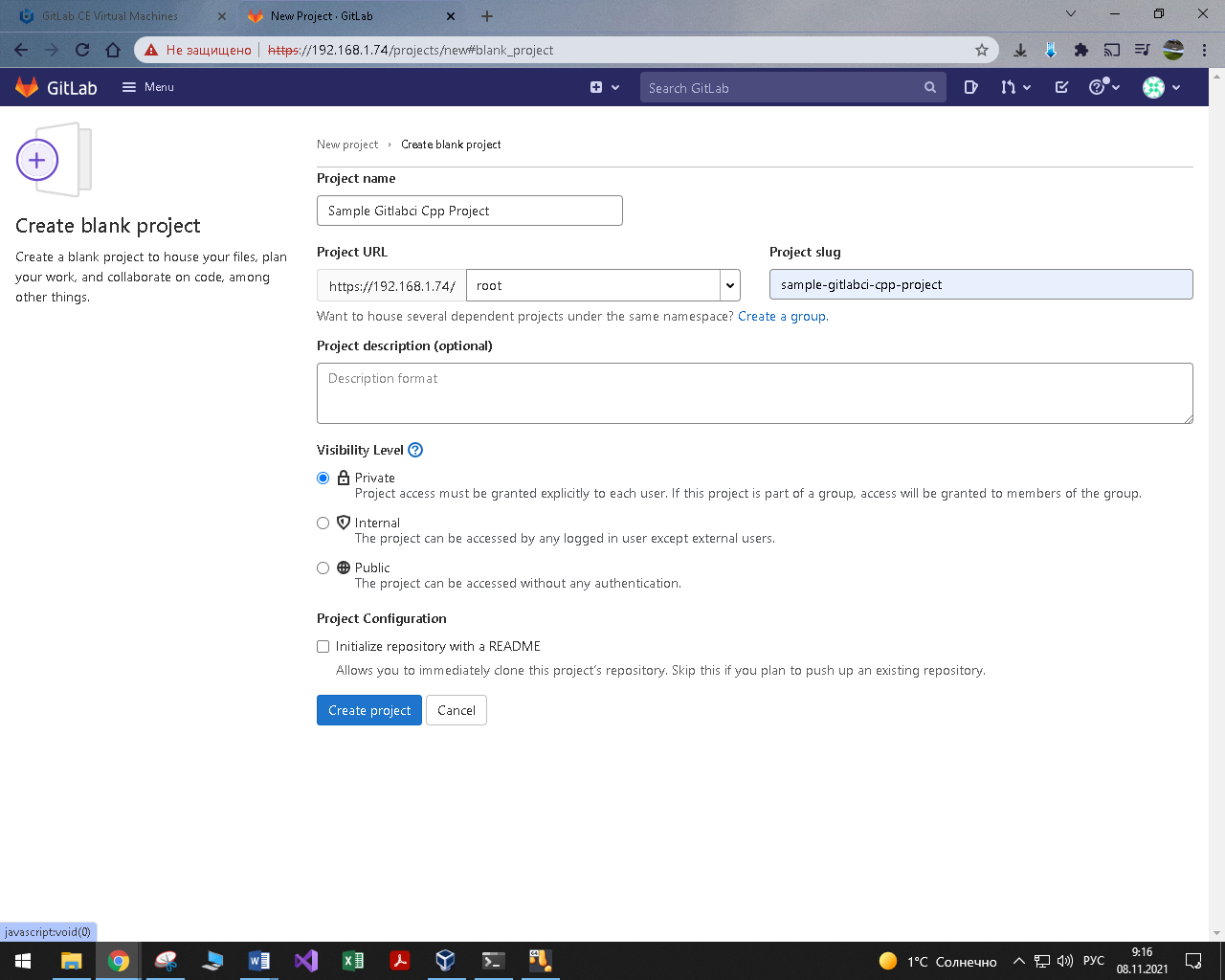


2.4.1 При создании проекта укажем следующие параметры:

Project slug: sample-gitlabci-cpp-project.

Visibility Level: Private.

Project Configuration: Initialize repository with a README (снять галочку, так как он Нам не нужен).



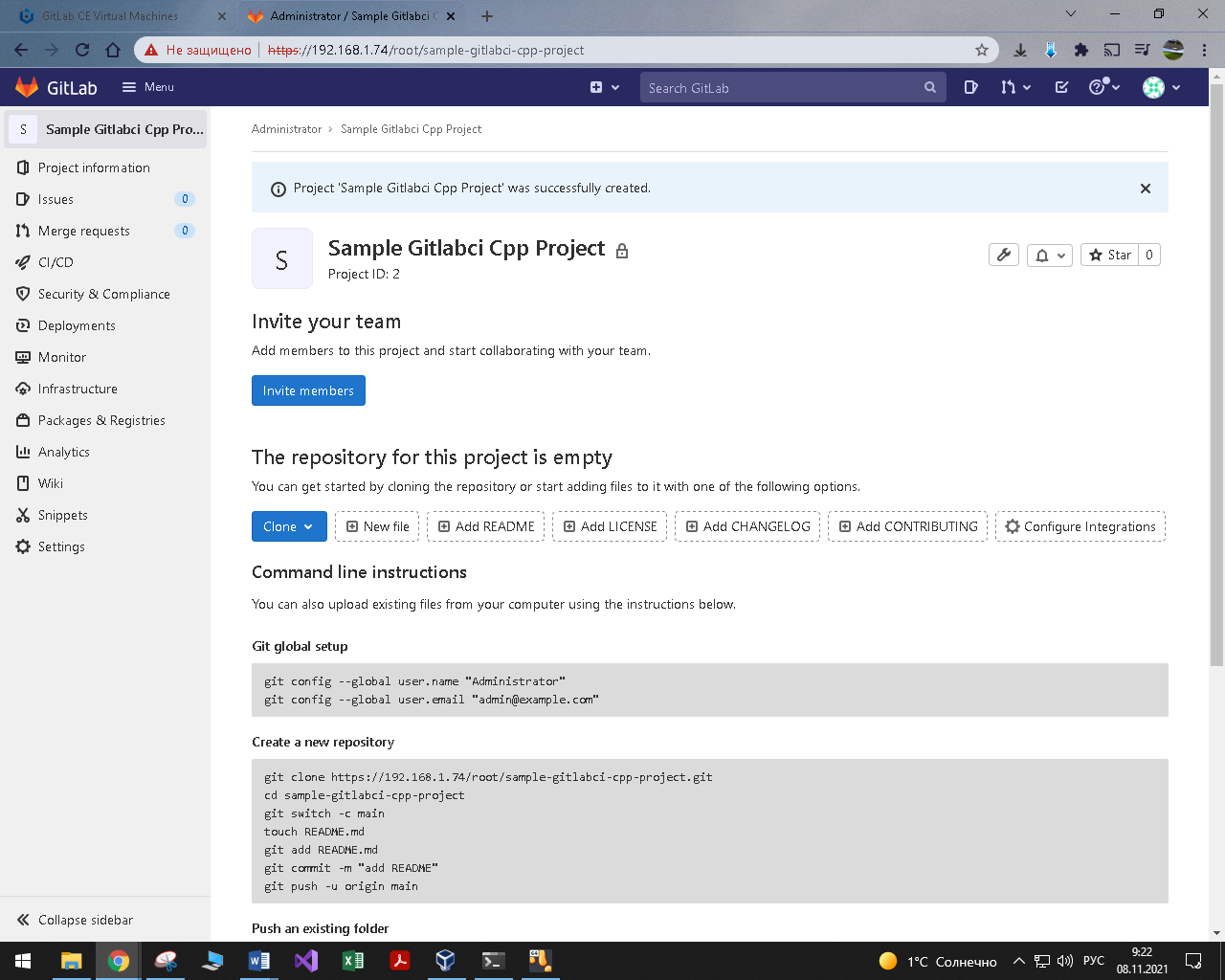
2.4.2 Репозиторий создан. На данном этапе так же выполним пару копирований, поскольку эти данные потребуются далее.

Данные для входа при работе с консолью git.

git config --global user.name "Administrator"

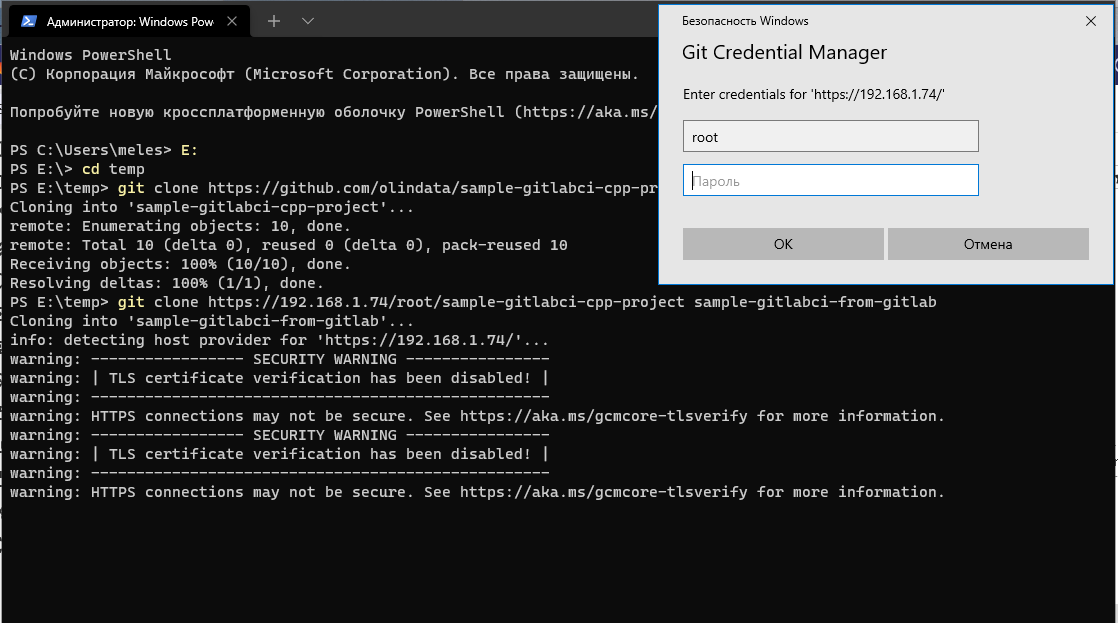
git config --global user.email [admin@example.com](mailto:admin@example.com)

А также регистрационный токен, для регистрации runner:   
zuaQ\_-QexbSvtHtFwJrr



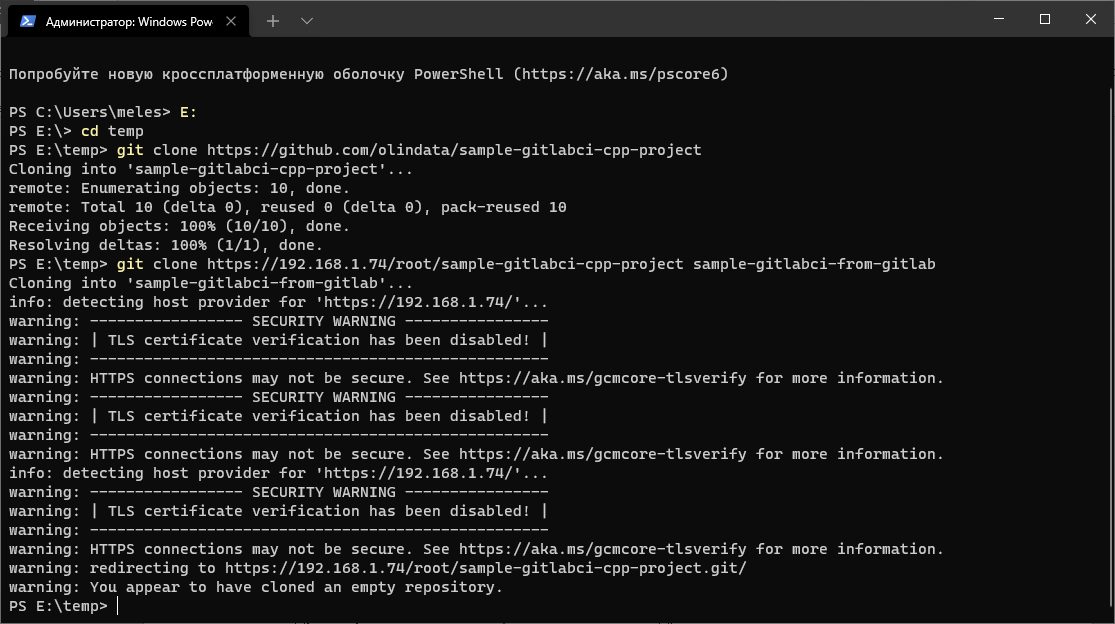
2.5 Клонируем, только что созданный репозиторий, к себе на ПК. Для этого перейдем в терминал и воспользуемся следующей командой:

git clone <https://192.168.1.74/root/sample-gitlabci-cpp-project> sample-gitlabci-from-gitlab

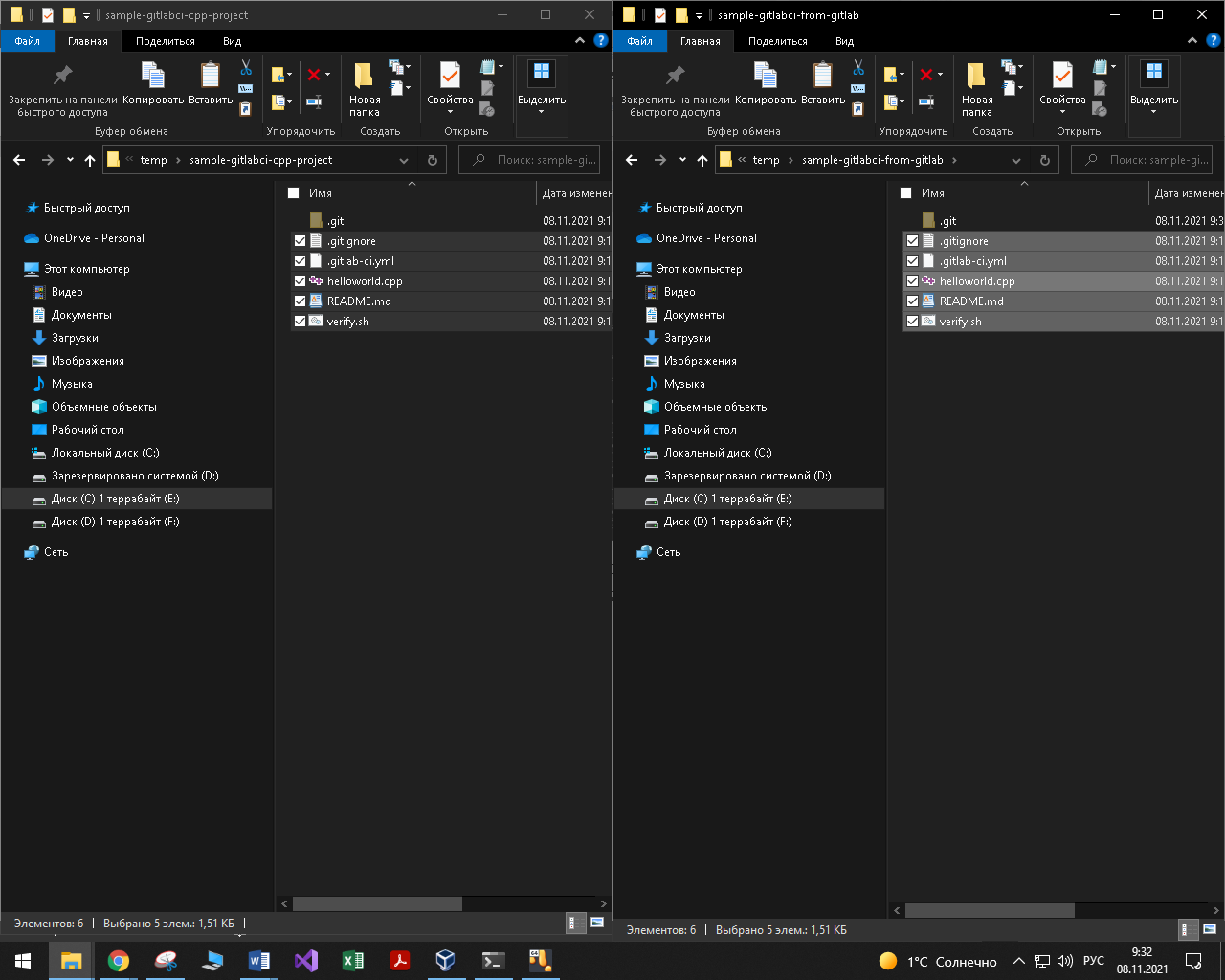


Так же на данном этапе необходимо указать логин и пароль, полученные на шаге 1, при выполнении действия 1.4.

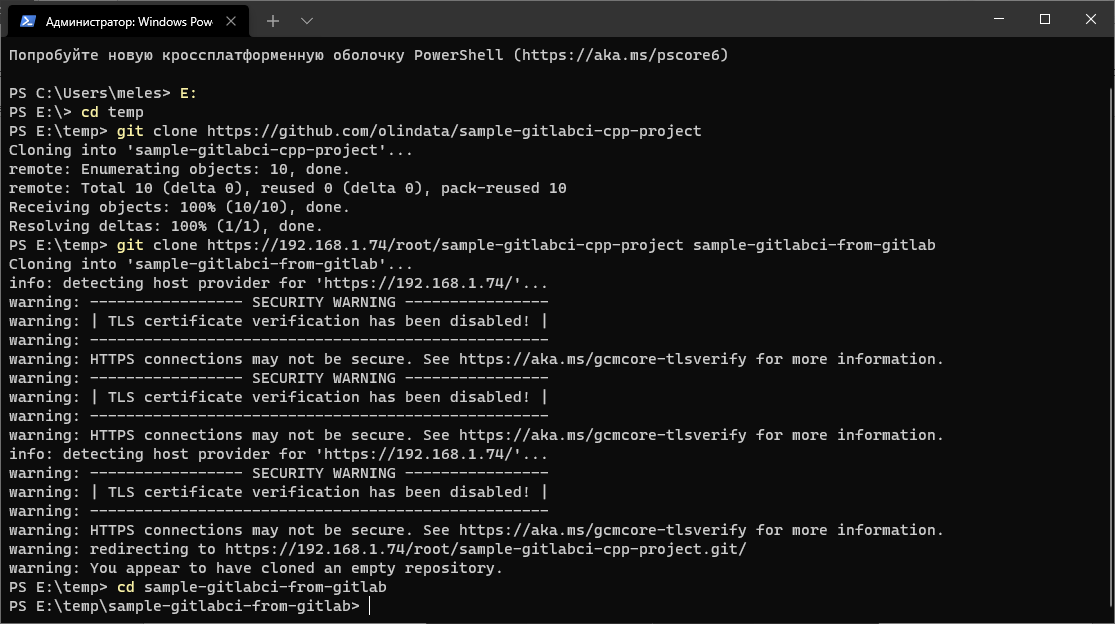
После чего репозиторий будет скопирован на ПК.



2.6 Для выгрузки проекта на сервер, нам потребуется скопировать файлы находящиеся в папке репозитория c Github в папку репозитория с Gitlab.



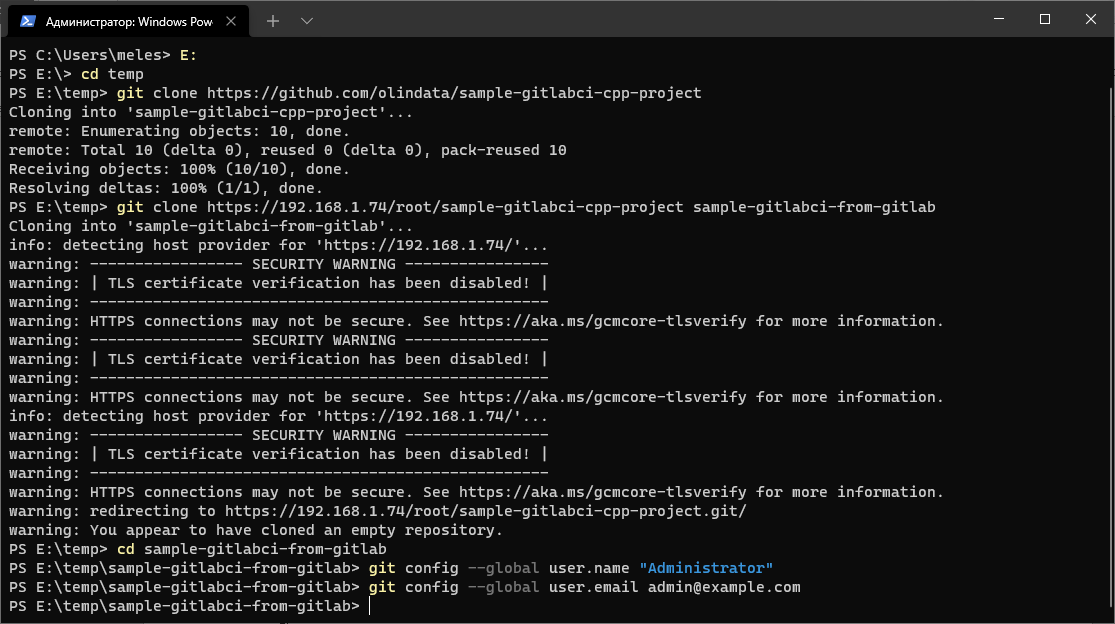
2.7 Перейдем в терминале в папку, в которую только что копировали данные и выполним выгрузку.



2.8 Перед началом выгрузки представимся системе, что это мы при помощи команд скопированных, при выполнении действия 2.4.2:

git config --global user.name "Administrator"

git config --global user.email [admin@example.com](mailto:admin@example.com)

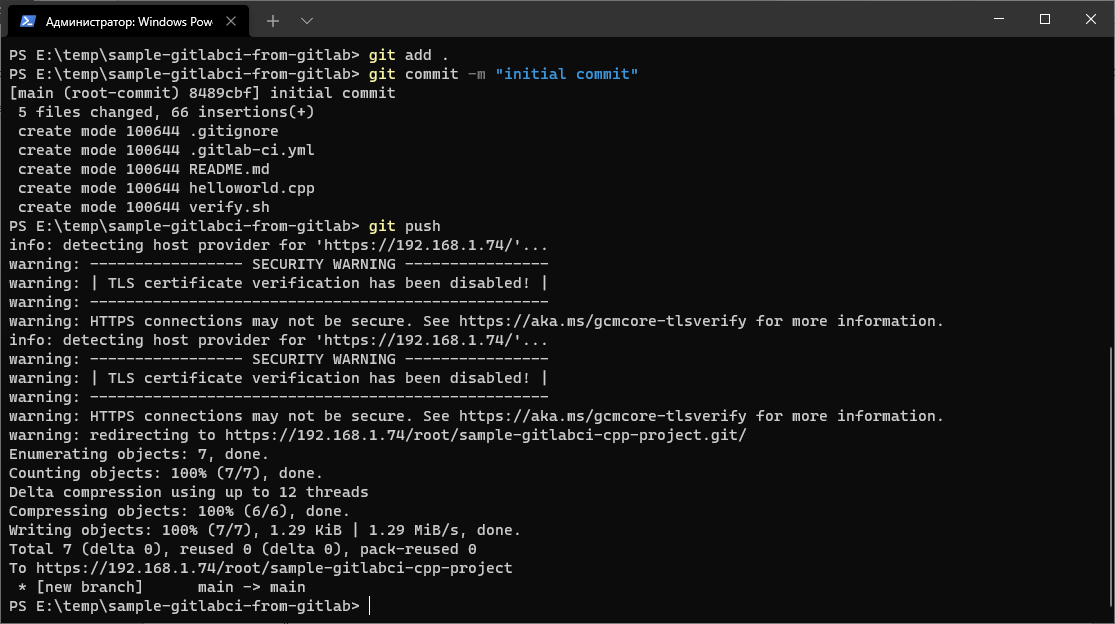


2.9 Выполним выгрузку на сервер. Для этого необходимо последовательно выполнить следующие команды:

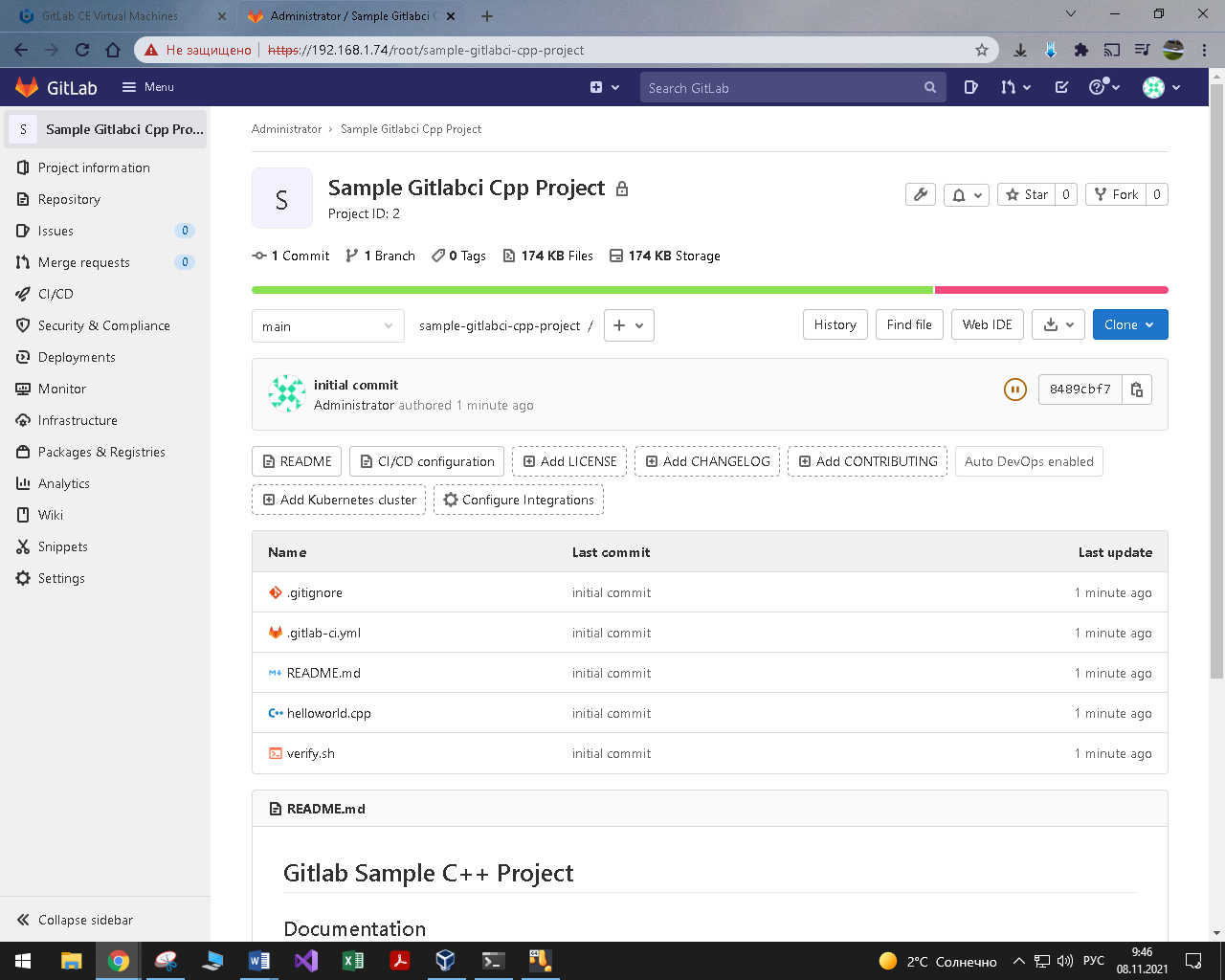
git add .

git commit –m “initial commit”

git push



2.10 Проверим выгрузку файлов на сервер.



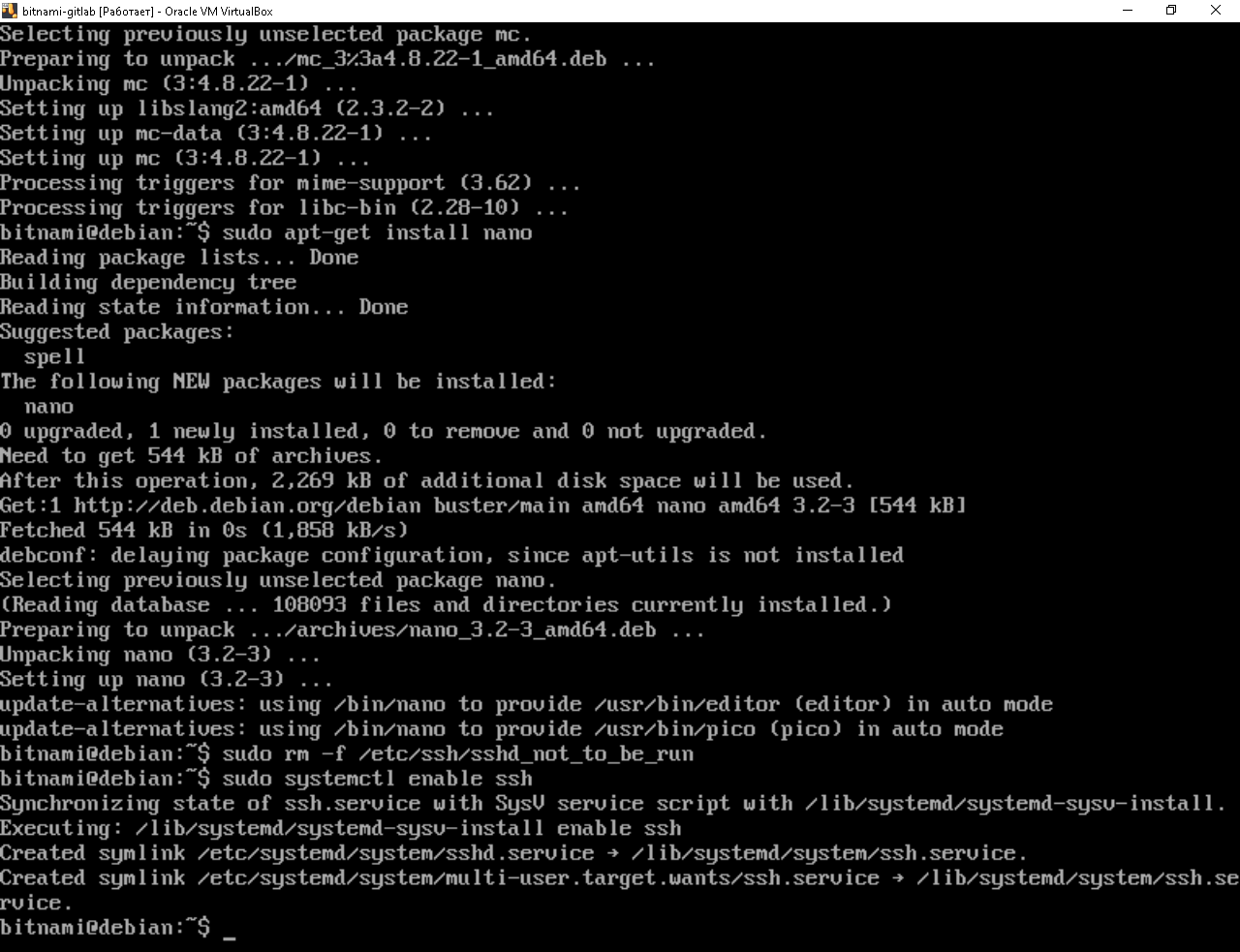
Шаг 3. Разблокируем SSH.

3.1 На сайте <https://docs.bitnami.com/virtual-machine/faq/get-started/enable-ssh/> представлены команды для этого:

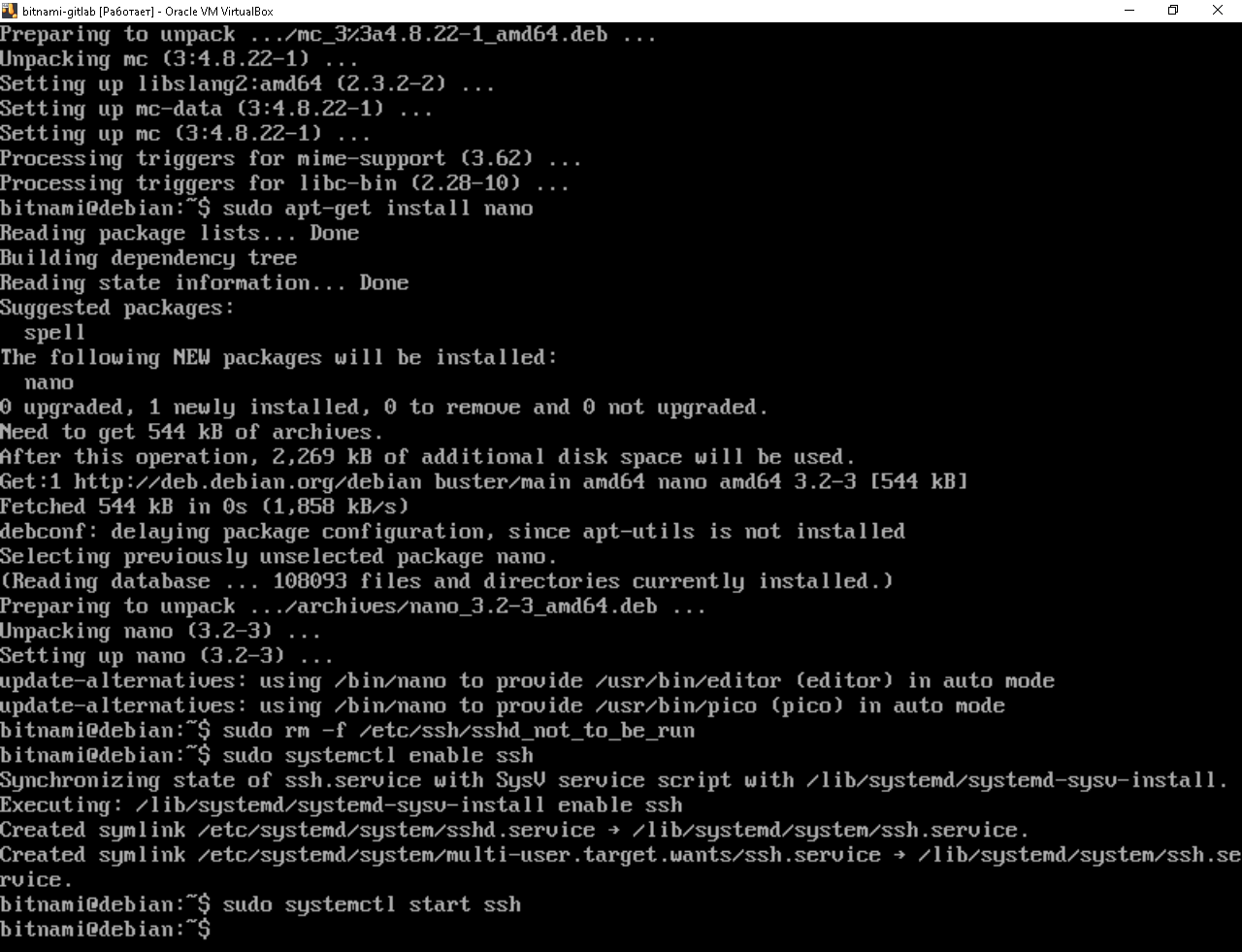
sudo rm -f /etc/ssh/sshd\_not\_to\_be\_run



sudo systemctl enable ssh

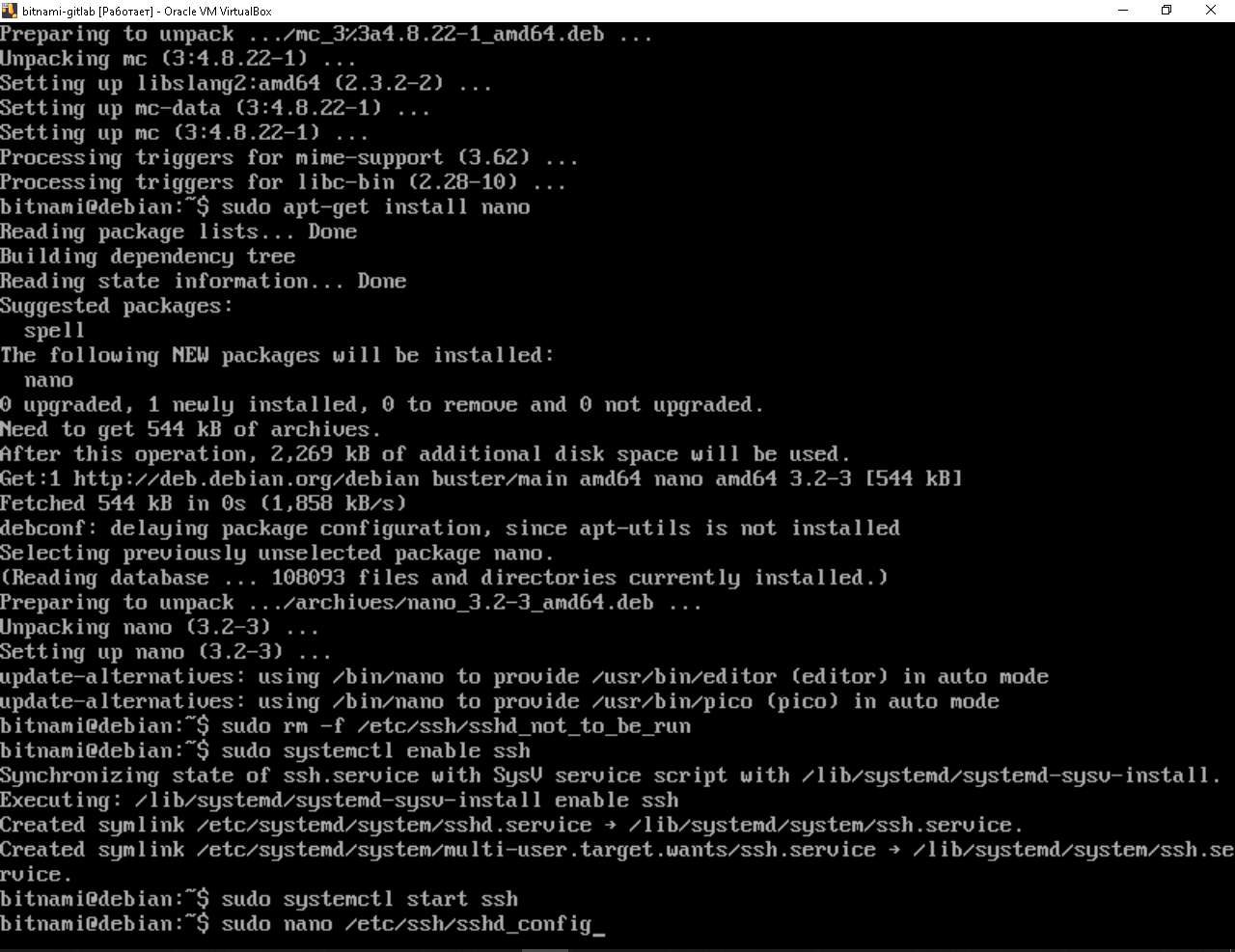


sudo systemctl start ssh

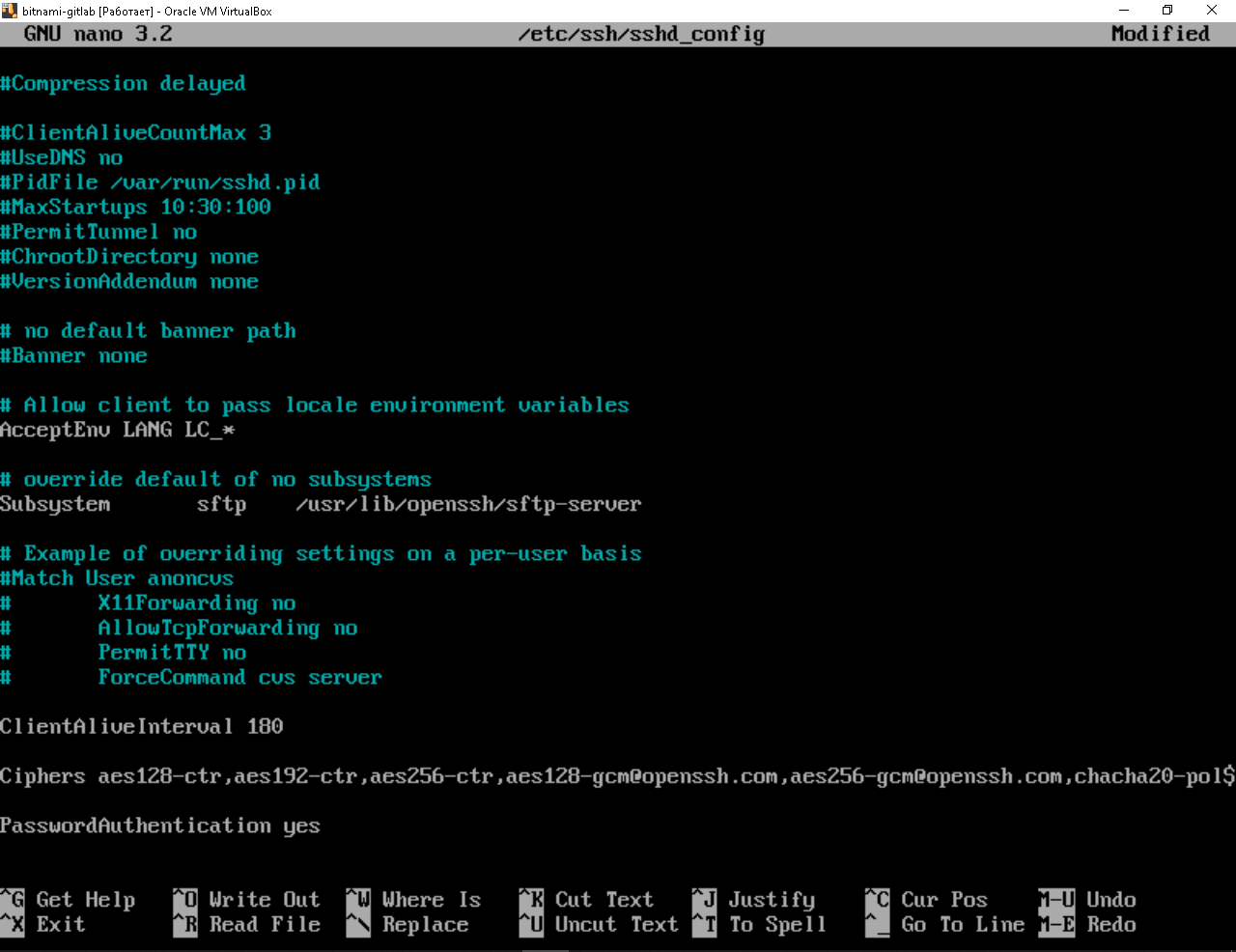


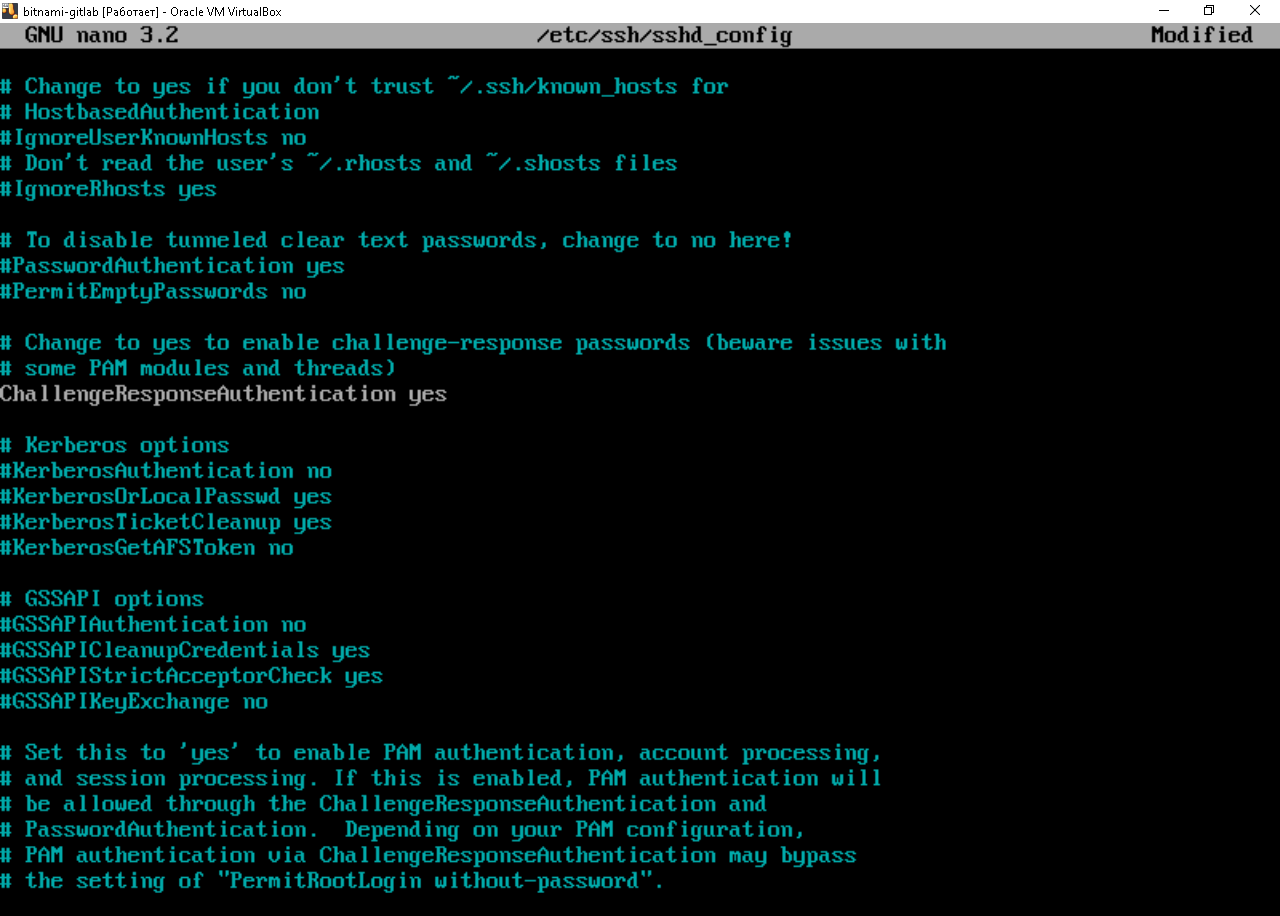
Шаг 4. SSH Public Key

4.1 Редактировать файл /etc/ssh/sshd\_config.



4.2 Изменить PasswordAuthentication и ChallengeResponseAuthentication на yes.

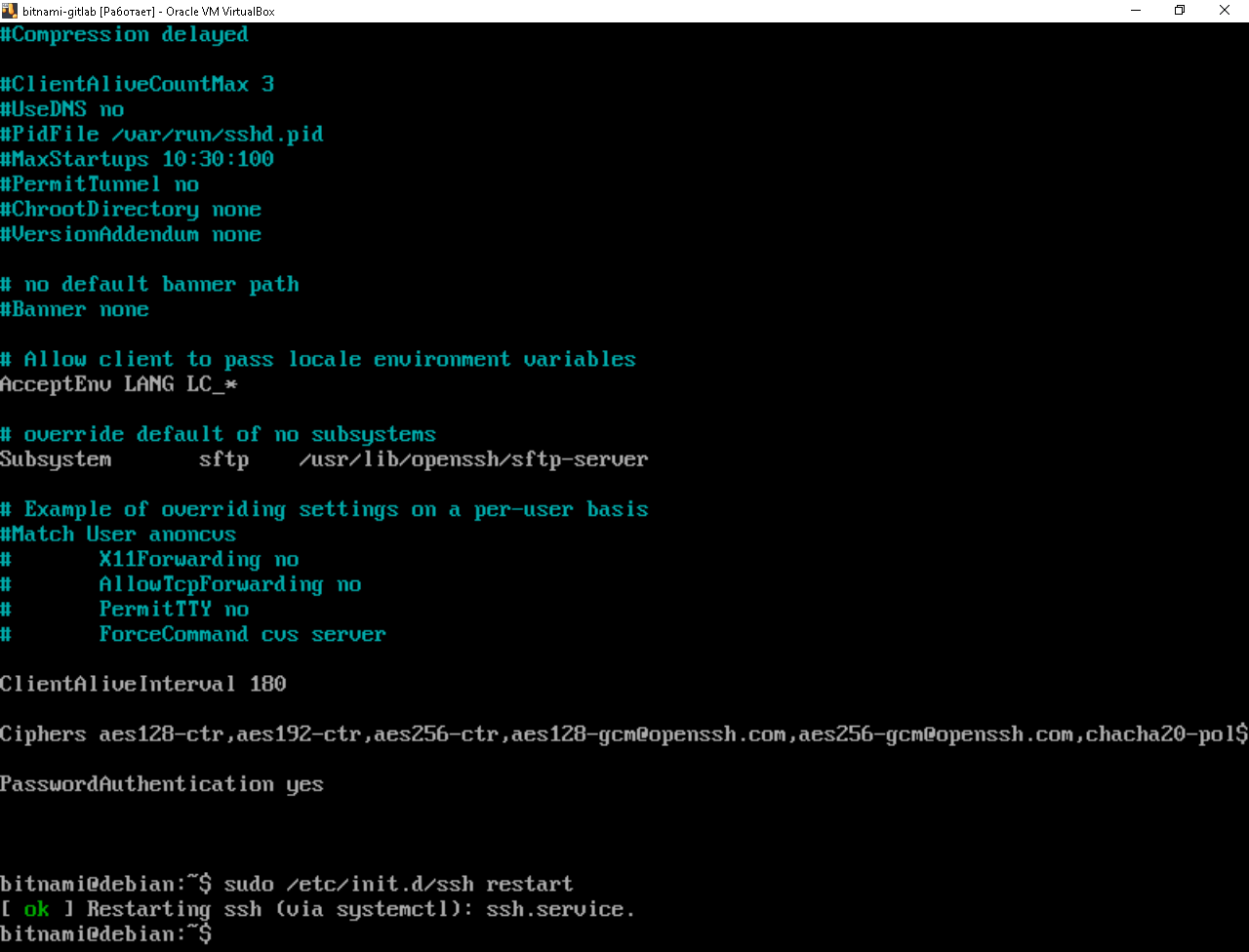




4.3 Перезапуск ssh-сервиса командой (на выбор)

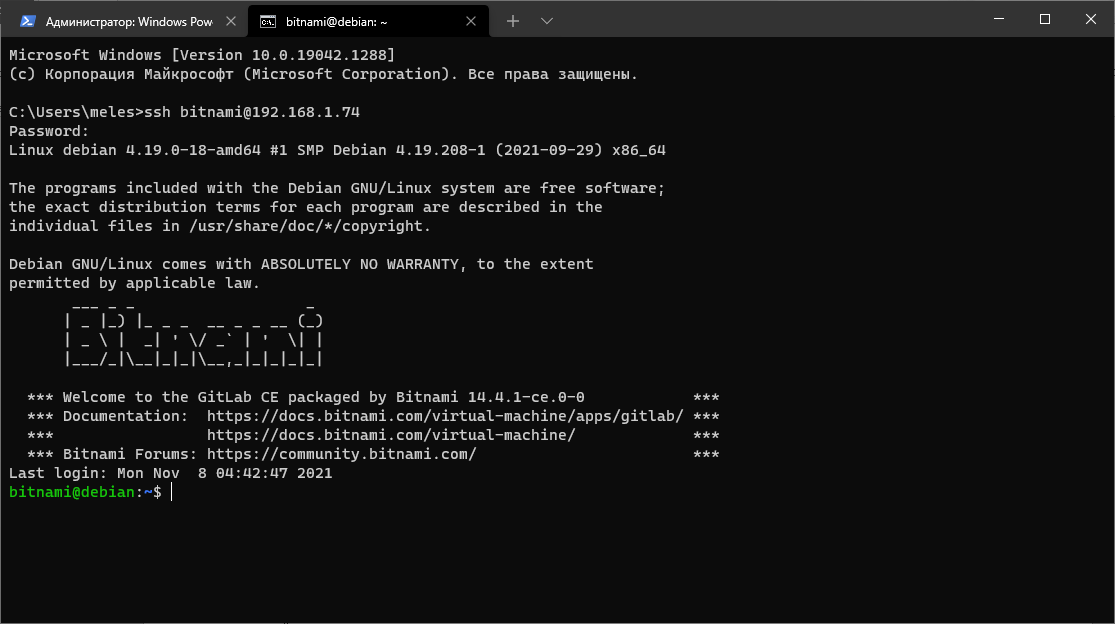
sudo /etc/init.d/ssh restart

sudo service sshd restart

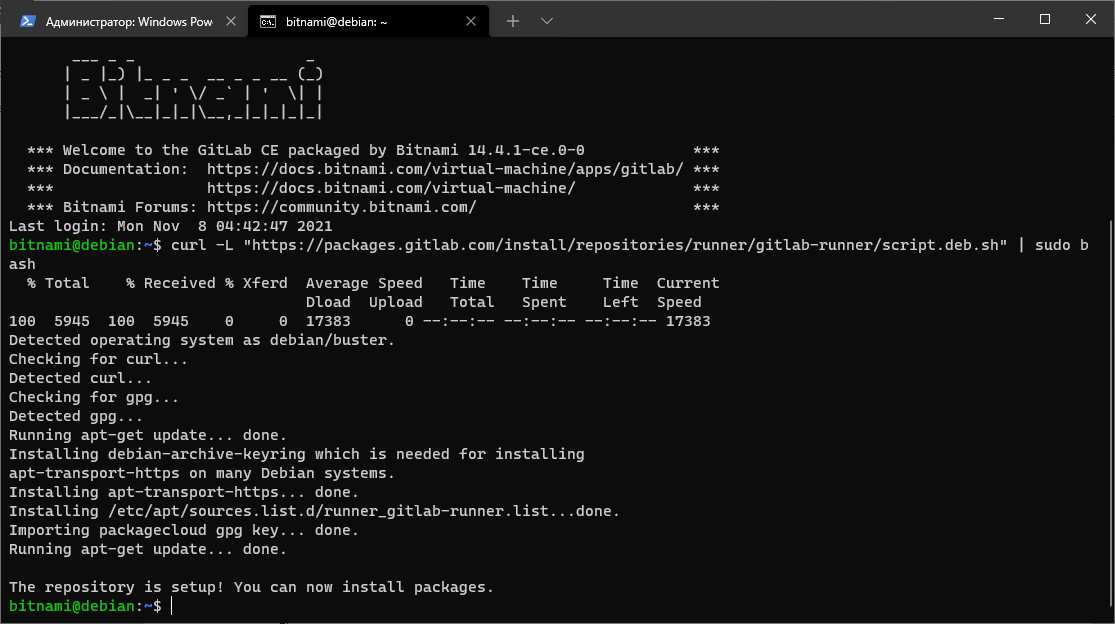


Шаг 5. Установка GitLab Runner.

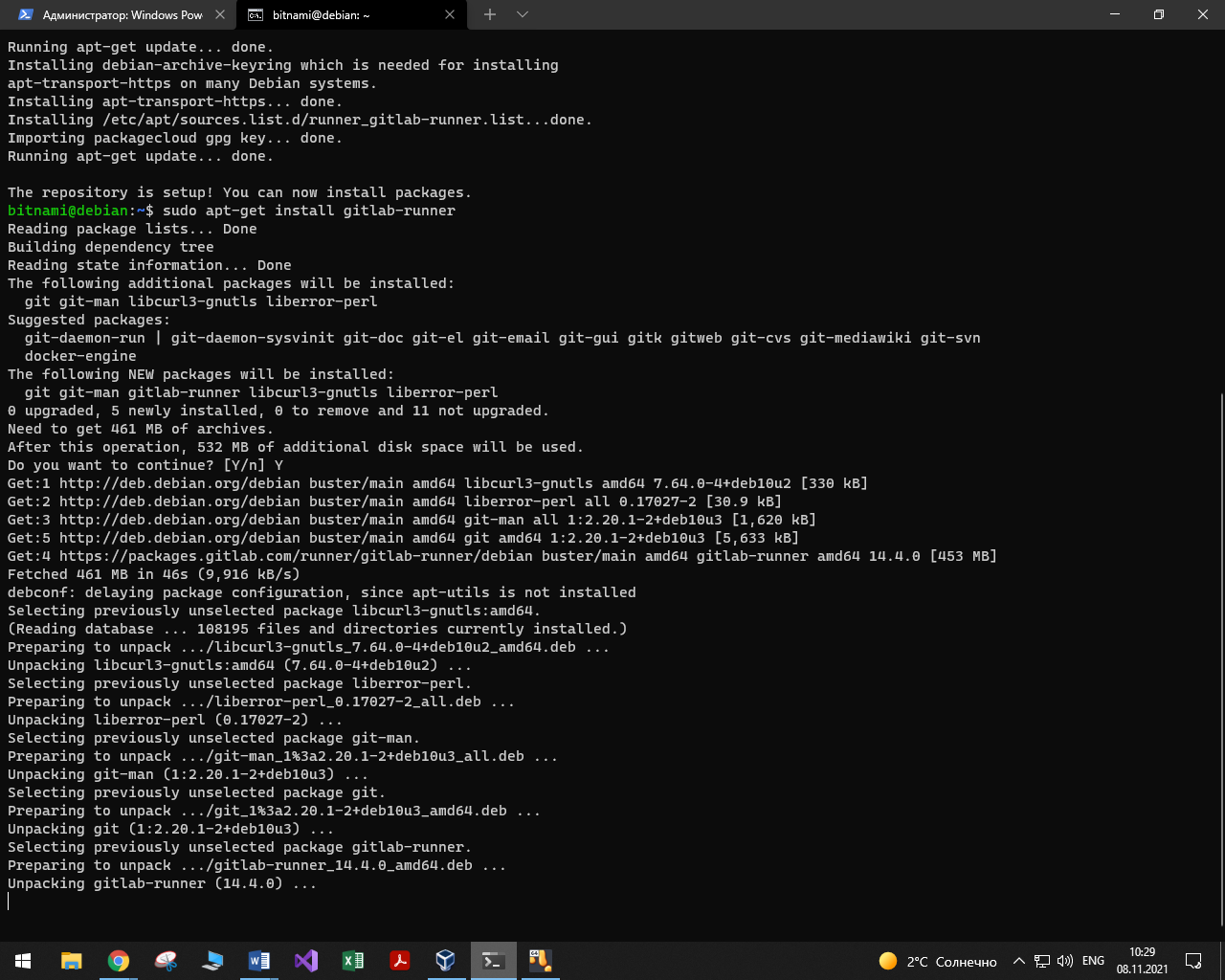
5.1 Перейдем в терминал и выполним подключение к виртуальной машине через SSH.



5.2 Для установки GitLab Runner добавим официальный репозиторий GitLab по ссылке: curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh" | sudo bash



5.3 Установка GitLab Runner

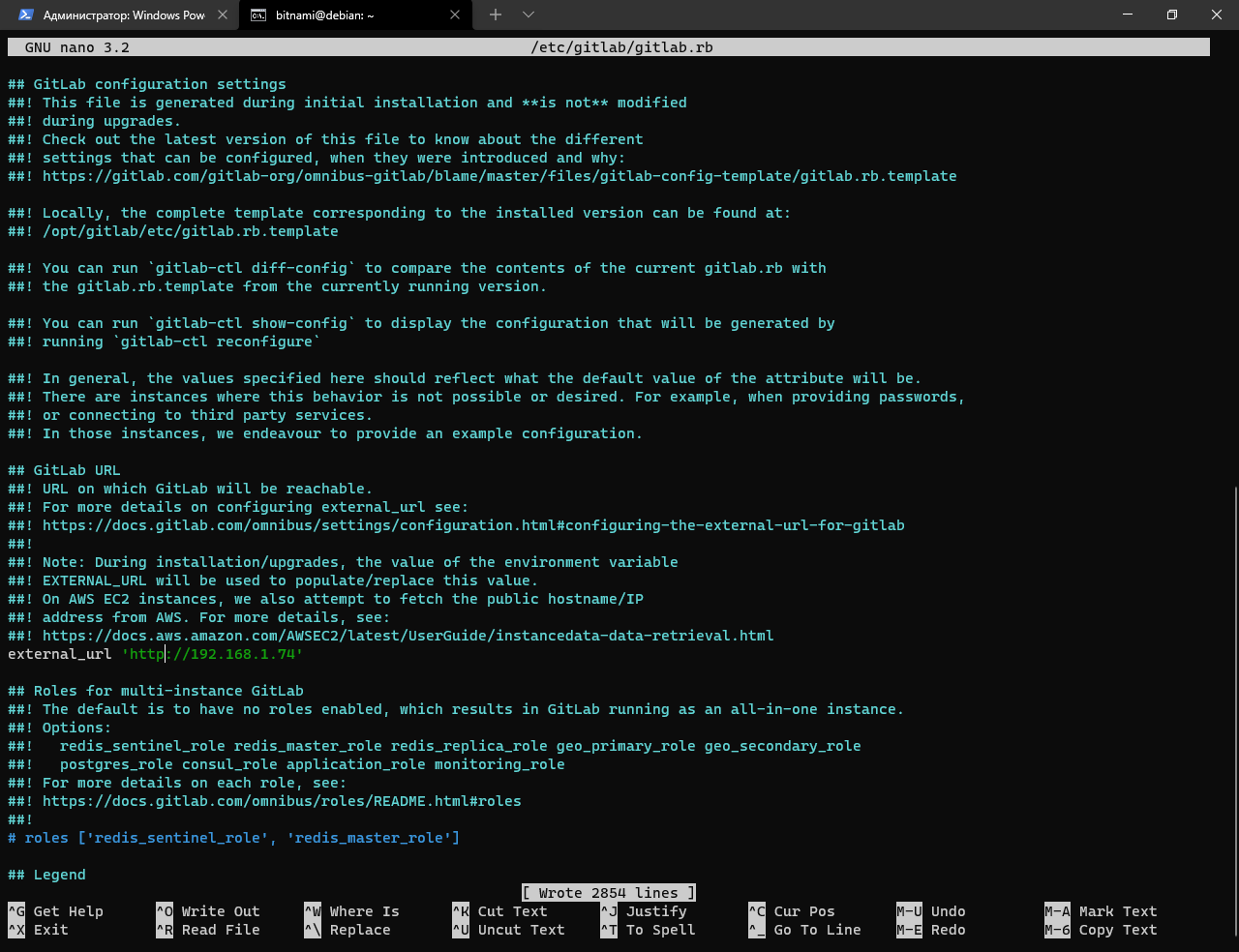


Шаг 6 Изменим /etc/gitlab/gitlab.rb для дезактивации https на gitlab

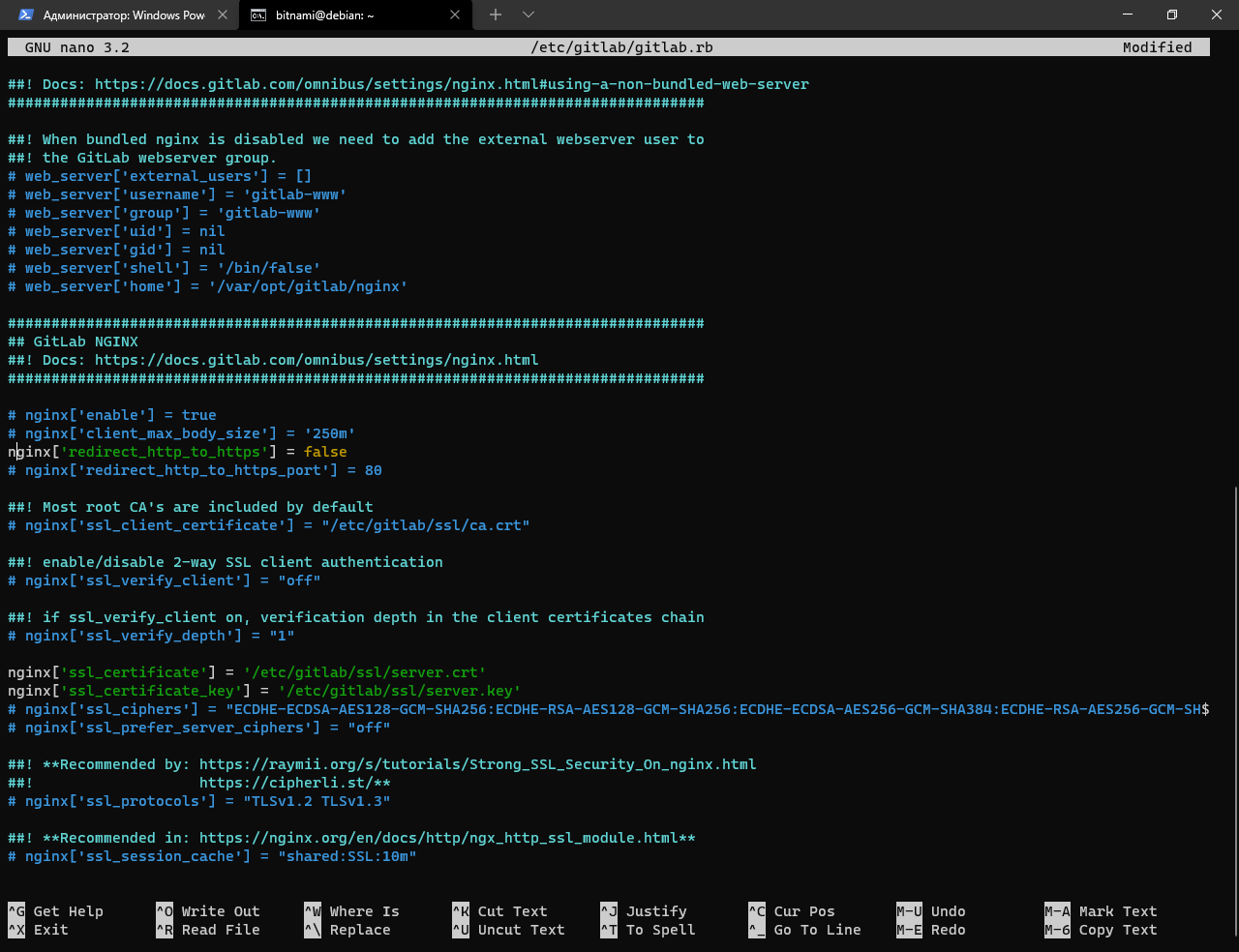
sudo nano /etc/gitlab/gitlab.rb



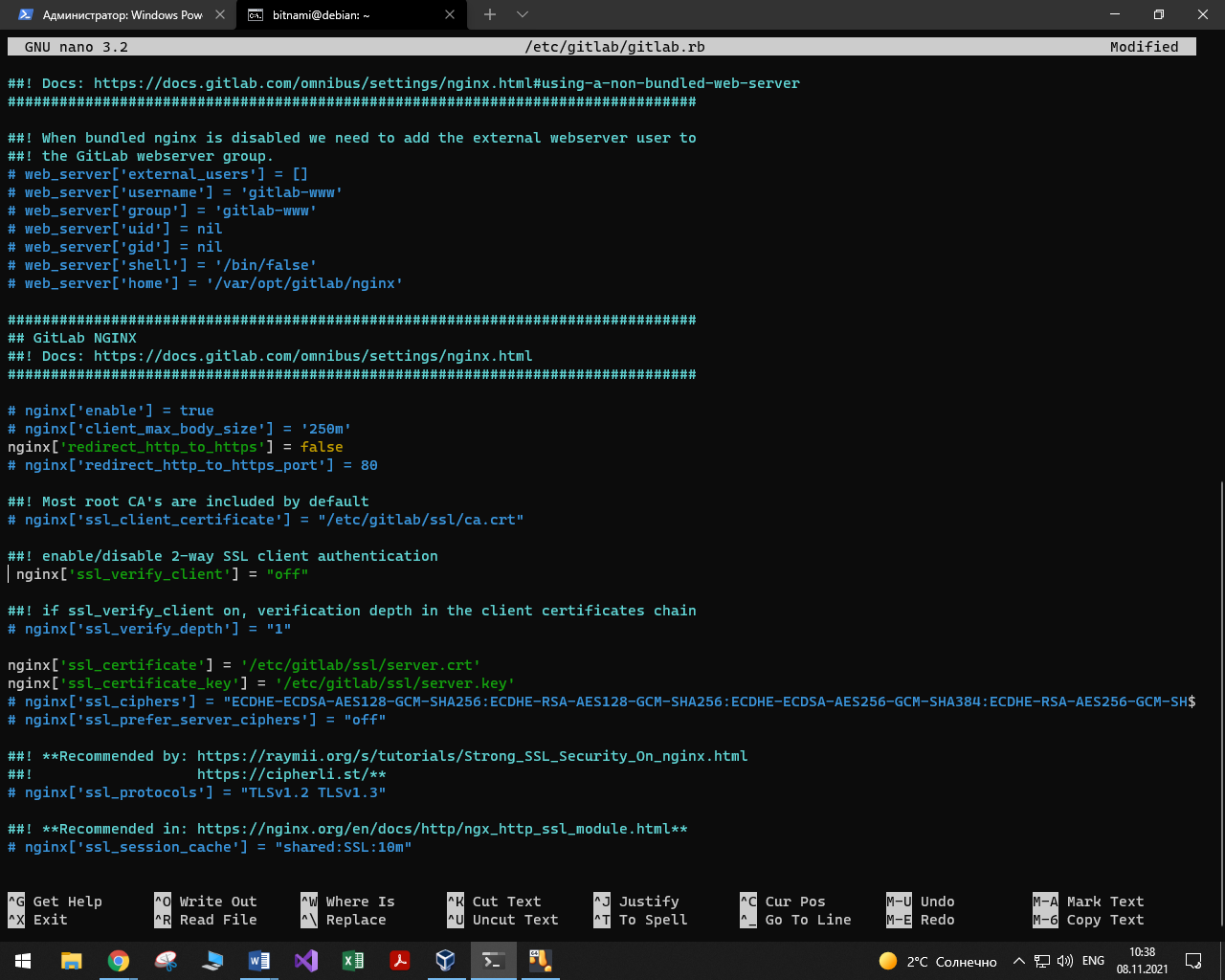
external\_url 'http://192.168.1.74



nginx['redirect\_http\_to\_https'] = false

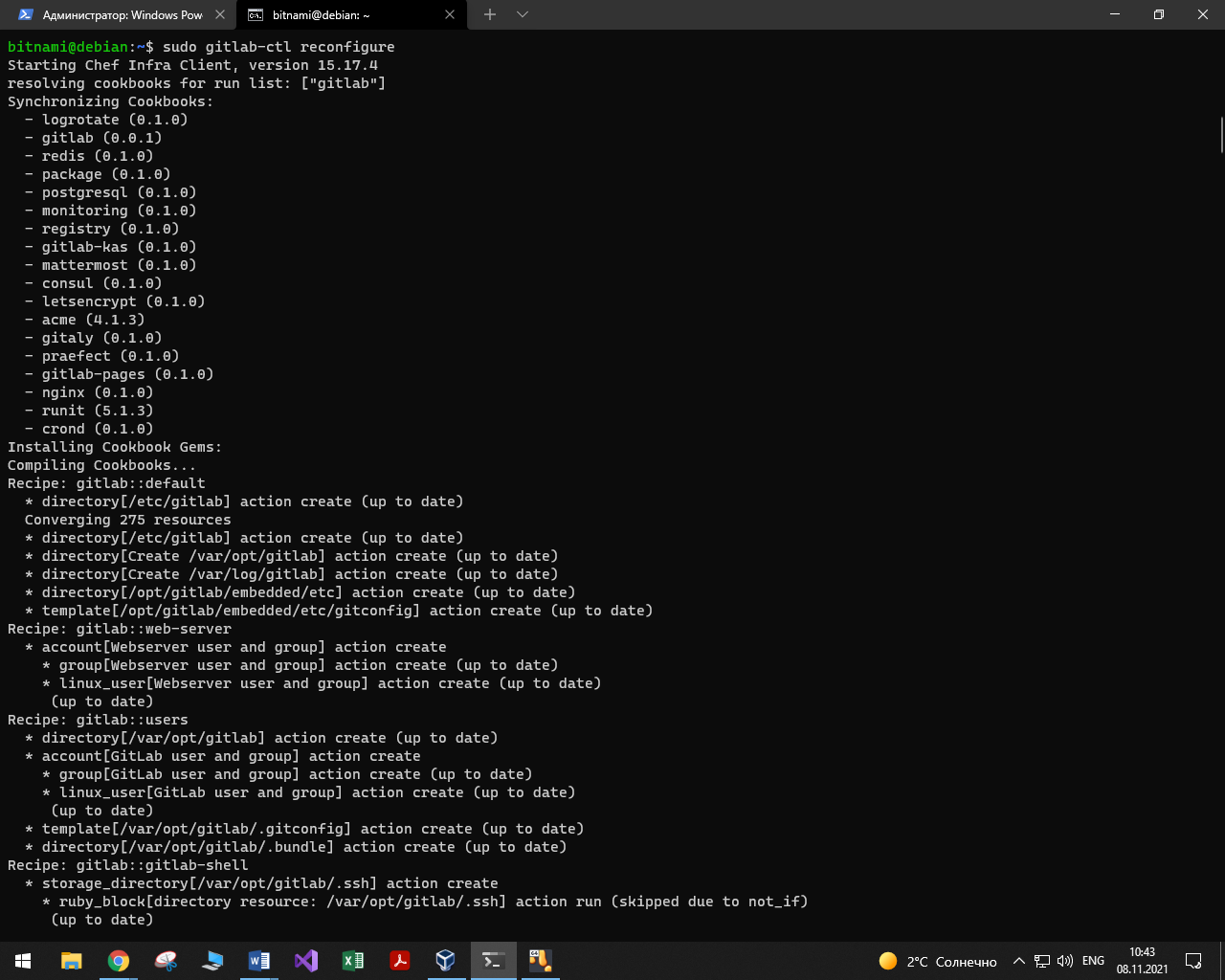


nginx['ssl\_verify\_client'] = "off"



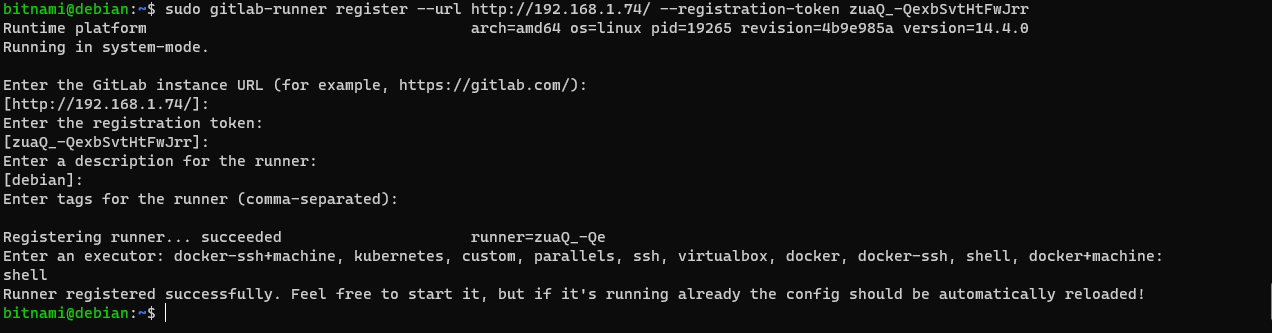
Шаг 7 Переконфигурируем GitLab, для того чтобы изменения вступили в силу:

7.1 Выполним команду sudo gitlab-ctl reconfigure



Шаг 8. Регистрация runner. Выбрать shell.

sudo gitlab-runner register --url http://192.168.1.74/ --registration-token zuaQ\_-QexbSvtHtFwJrr

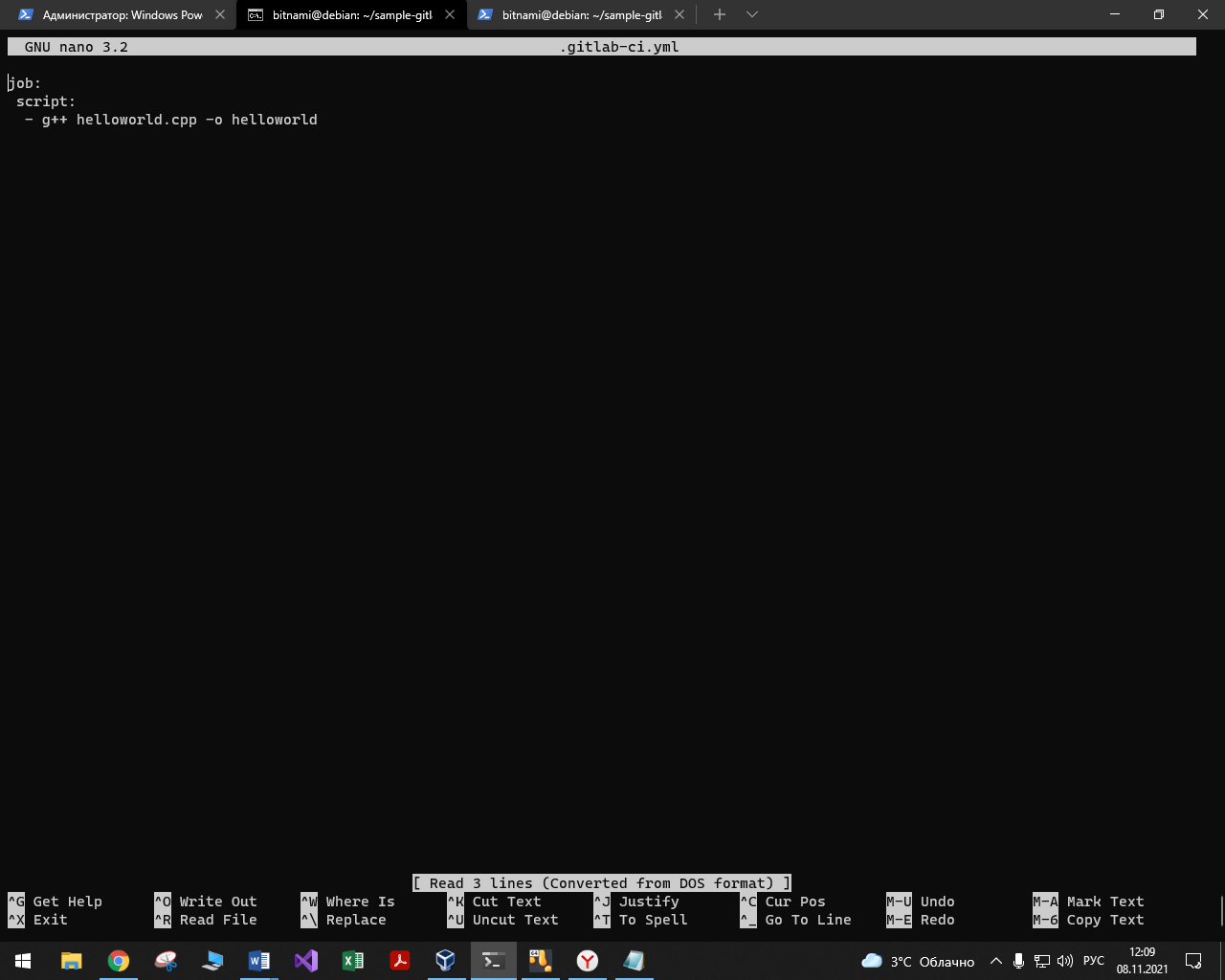


Шаг 9 Изменить .gitlab-ci.yml для запуска runner в режиме shell (без Docker)

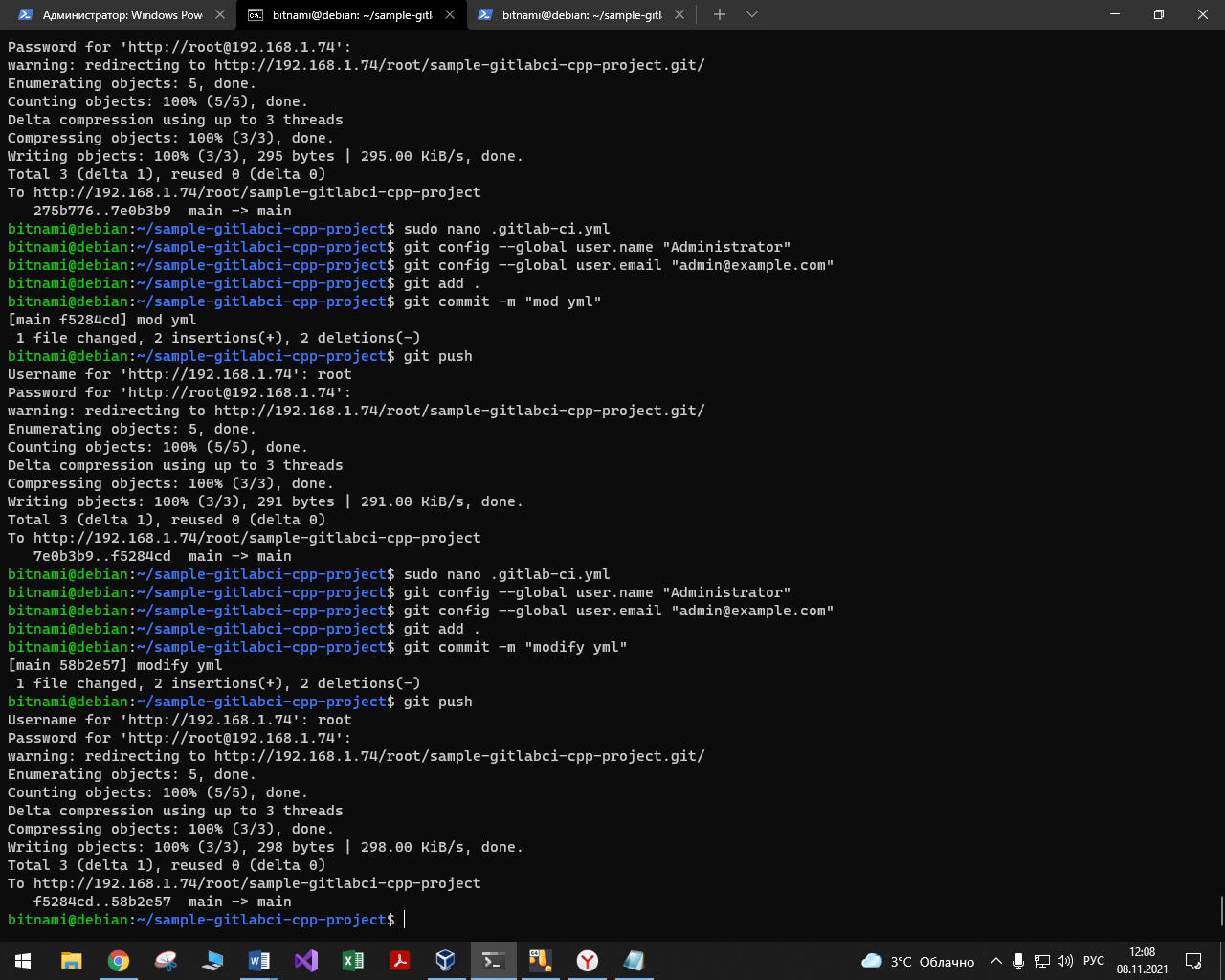
9.1 Перейдем в папку проекта.



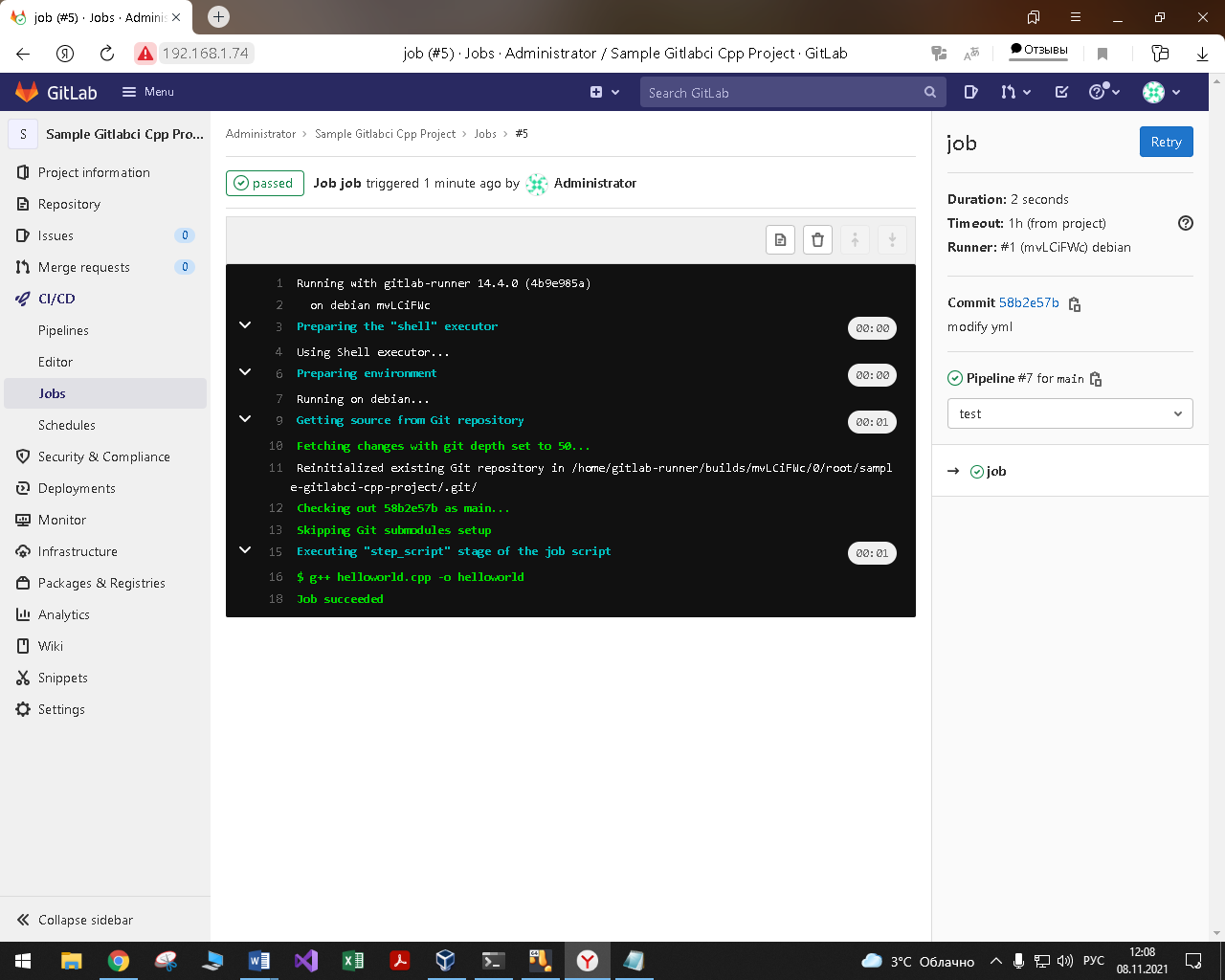
9.2 Отредактируем .gitlab-ci.yml



9.3 Запушим введенные изменения.



Шаг 10 Запустить Pipeline: CI/CD > Pipelines > Run pipeline



**CONCLUSION**

In the conclusion of the laboratory work, the basic utilities of Git and CI were studied for working with repositories, runners and pipelines; all steps completed successfully.