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: «Развёртывание и жизненный цикл программного обеспечения»

Тема: **«Git and CI»**

Выполнил: Доружинский Д.В.

Группа: 6133-010402D

Самара 2021

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc92294913)

[ЗАДАНИЕ 4](#_Toc92294914)

[ВВЕДЕНИЕ 6](#_Toc92294915)

[1 Загрузка и установка образа виртуальной машины Gitlab-Bitnami. 7](#_Toc92294916)

[1.1 Скачаем образ виртуальной машины с официального сайта *https://bitnami.com/stack/gitlab/virtual-machine.* 7](#_Toc92294917)

[1.2 Импортируем образ в Oracle VM VirtualBox*.* 7](#_Toc92294918)

[1.3 Настройка виртуальной машины. 8](#_Toc92294919)

[1.4 Запуск настроенной виртуальной машины. 9](#_Toc92294920)

[1.5 Вход на сервер приложения 9](#_Toc92294921)

[1.6 Вход в приложение 10](#_Toc92294922)

[2 Загрузка проекта с *https://github.com/olindata/sample-gitlabci-cpp-project на наш сервер Gitlab*. 11](#_Toc92294923)

[2.1 Установим 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* 11](#_Toc92294924)

[2.2 После установки git-scm откроем терминал и создадим пустую папку, в которую будем клонировать репозиторий с сайта *https://github.com/olindata/sample-gitlabci-cpp-project* 11](#_Toc92294925)

[2.3 Клонируем репозиторий в эту папку, при помощи команды: 11](#_Toc92294926)

[2.4 Создадим пустой репозиторий в приложении на сервере. 12](#_Toc92294927)

[2.5 Клонируем созданный репозиторий к себе на ПК. 13](#_Toc92294928)

[2.6 Для выгрузки проекта на сервер, необходимо скопировать файлы, находящиеся в папке репозитория с GitHub, в папку репозитория GitLab 14](#_Toc92294929)

[2.7 Перейдем через терминал в папку репозитория GitLab и выполним выгрузку на сервер. 14](#_Toc92294930)

[2.8 Проверим выгрузку файлов на сервер 15](#_Toc92294931)

[3 Разблокируем SSH 16](#_Toc92294932)

# ЗАДАНИЕ

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

In Job Logs you should see something like this:

*Running with gitlab-runner 14.3.2 (e0218c92)*

*on debian Zv4xR9-x*

*Preparing the "shell" executor*

*00:00*

*Using Shell executor...*

*Preparing environment*

*00:00*

*Running on debian...*

*Getting source from Git repository*

*00:00*

*Fetching changes with git depth set to 50...*

*Reinitialized existing Git repository in /home/gitlab-runner/builds/Zv4xR9-x/0/root/sample-gitlabci-cpp-project/.git/*

*Checking out 69a8ab3b as main...*

*Removing helloworld*

*Skipping Git submodules setup*

*Executing "step\_script" stage of the job script*

*00:01*

*$ g++ helloworld.cpp -o helloworld*

*Job succeeded*

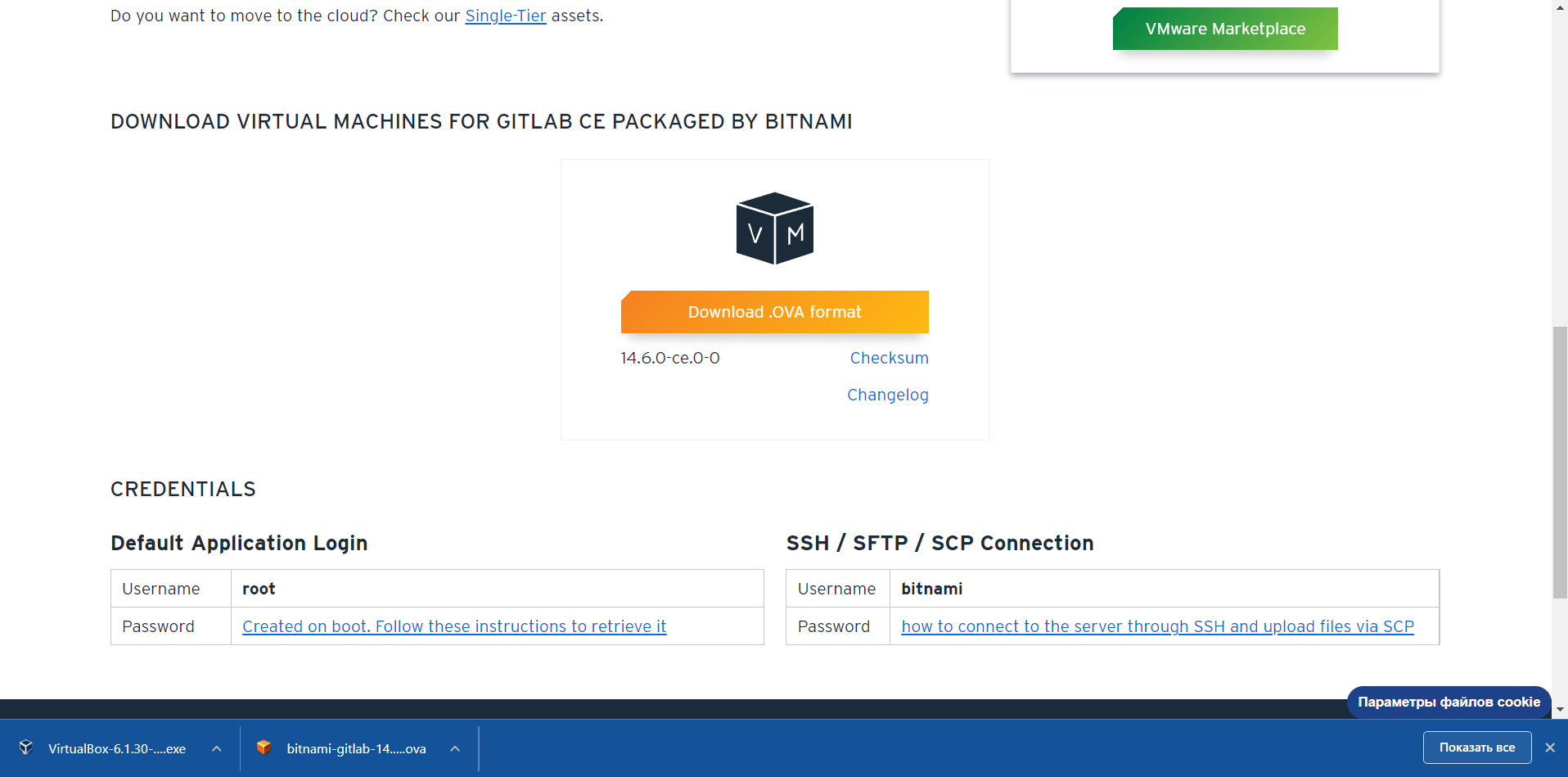
# ВВЕДЕНИЕ

Ход лабораторной работы “Git and CI”:

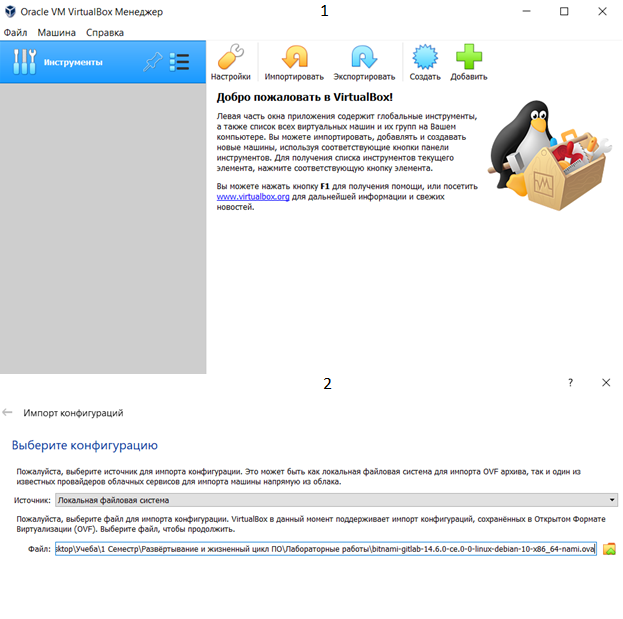
1. Загрузка и установка образа виртуальной машины Gitlab-Bitnami.
2. Загрузка проекта с *https://github.com/olindata/sample-gitlabci-cpp-project* на наш сервер Gitlab.
3. Разблокируем SSH.
4. SSH Public Key.
5. Установка GitLab Runner.
6. Изменим /etc/gitlab/gitlab.rb для дезактивации https на gitlab sudo nano /etc/gitlab/gitlab.rb.
7. Переконфигурируем GitLab, для того чтобы изменения вступили в силу.
8. Регистрация runner. Выбрать shell.
9. Изменить .gitlab-ci.yml для запуска runner в режиме shell (без Docker).
10. Запустить Pipeline: CI/CD > Pipelines > Run pipeline

## Загрузка и установка образа виртуальной машины Gitlab-Bitnami.

### Скачаем образ виртуальной машины с официального сайта [*https://bitnami.com/stack/gitlab/virtual-machine*](https://bitnami.com/stack/gitlab/virtual-machine)*.*



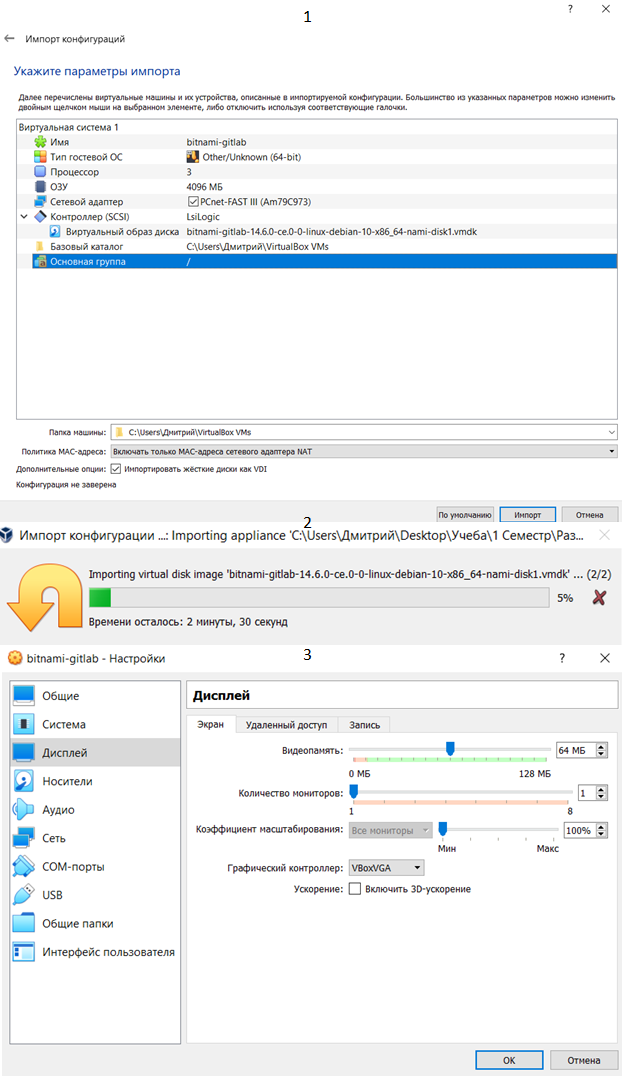
### Импортируем образ в Oracle VM VirtualBox*.*



### Настройка виртуальной машины.

Параметры:

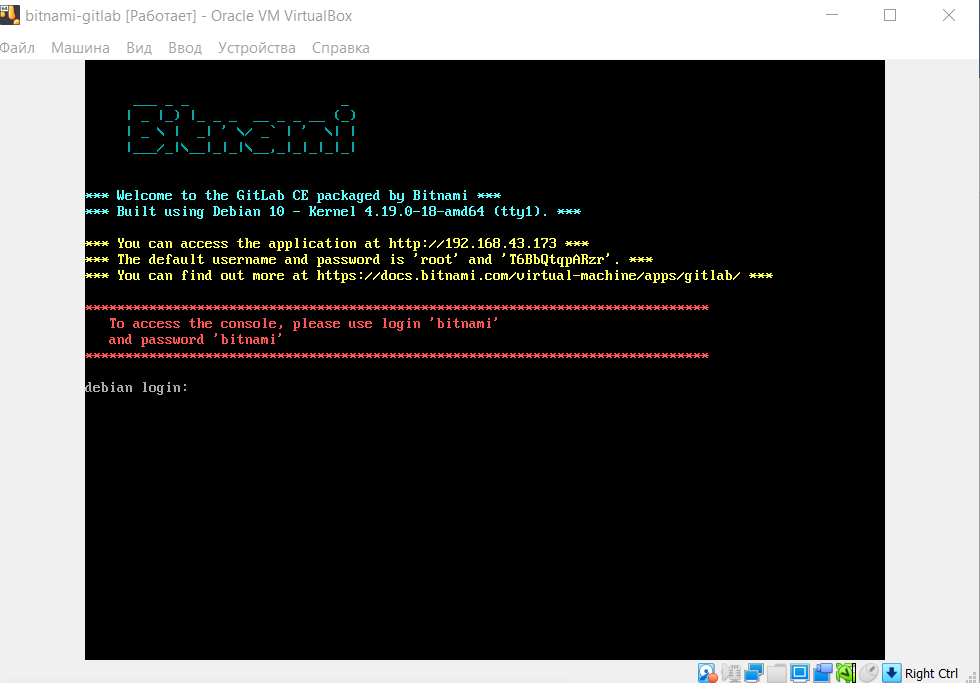
* физических ядра процессора
* 4096 Мб оперативной памяти
* 64 Мб видеопамяти



### Запуск настроенной виртуальной машины.

На данном этапе необходимо записать следующие данные:

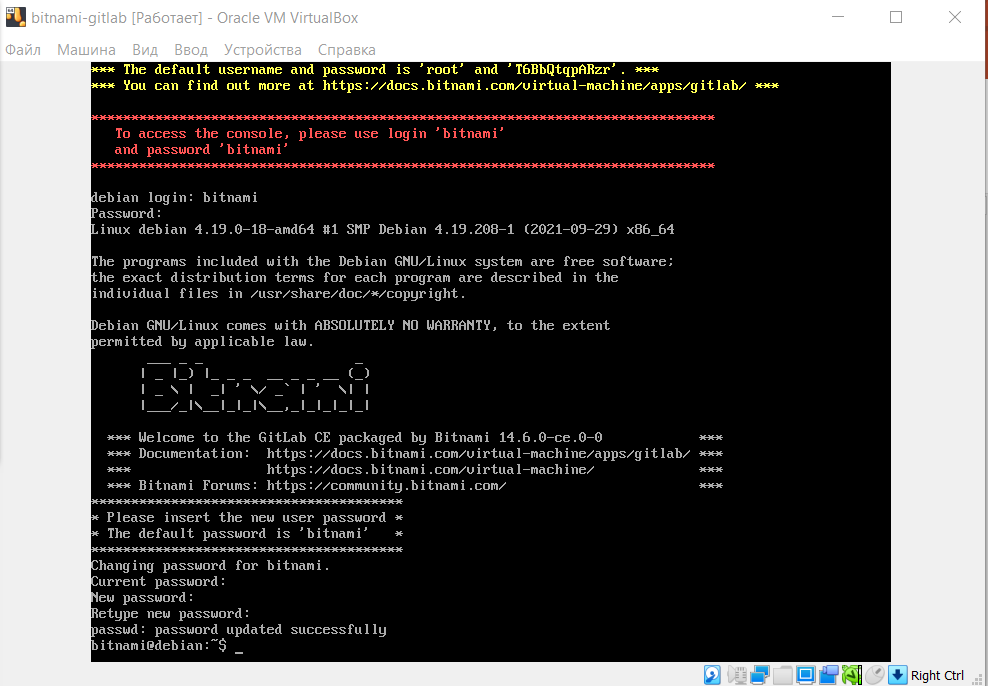
* Адрес приложения - **https://192.168.43.173**;
* Username – **root**;
* Password - **T6BbQtqpARzr**.



### Вход на сервер приложения

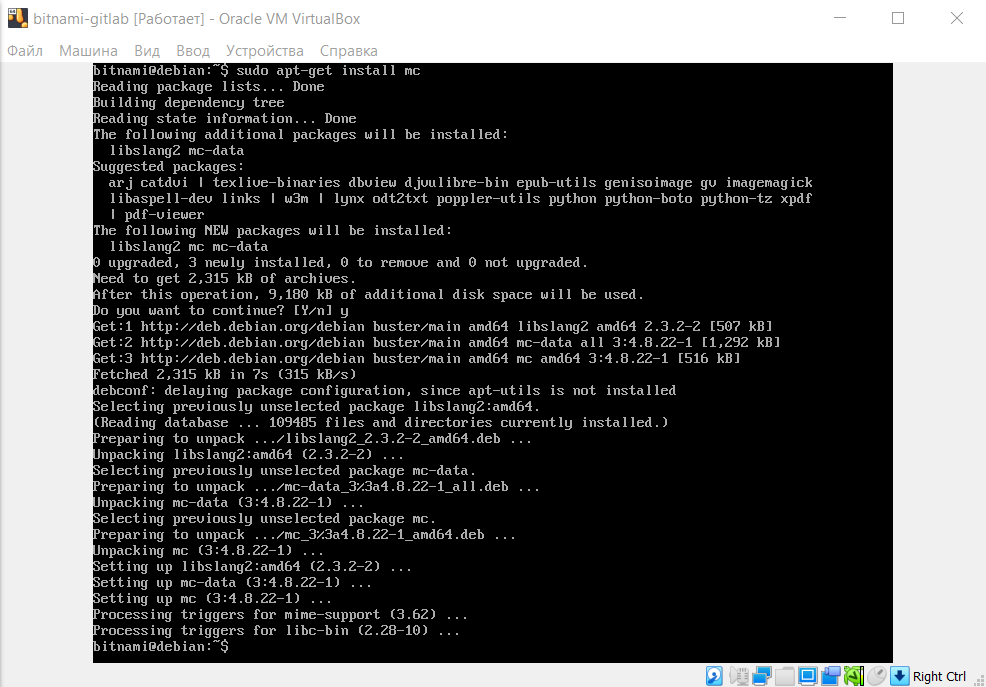
По умолчанию login и password для входа на сервер являются **bitnami** и **bitnami**. После первого входа, необходимо ввести старый password и задать двумя повторами. После необходимо подождать для полного развёртывания и инициализации виртуальной машины.

Вводим старый пароль и задаём новый двумя повторами.



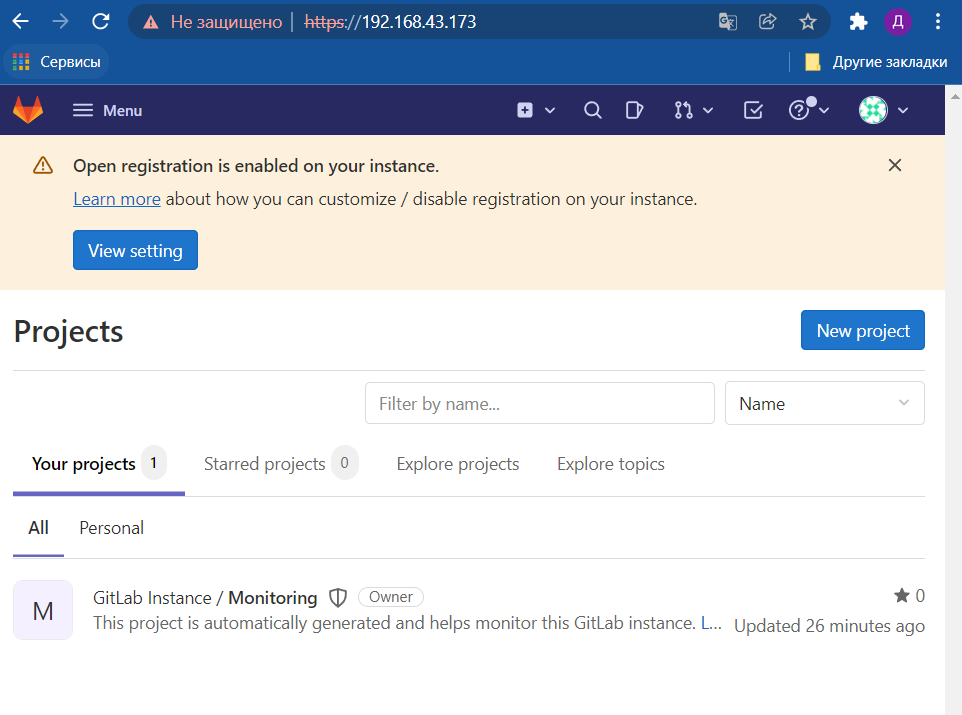
Установим Total Commander и аналог текстового редактора следующими командами:

* sudo apt-get install mc;
* sudo apt get install nano.



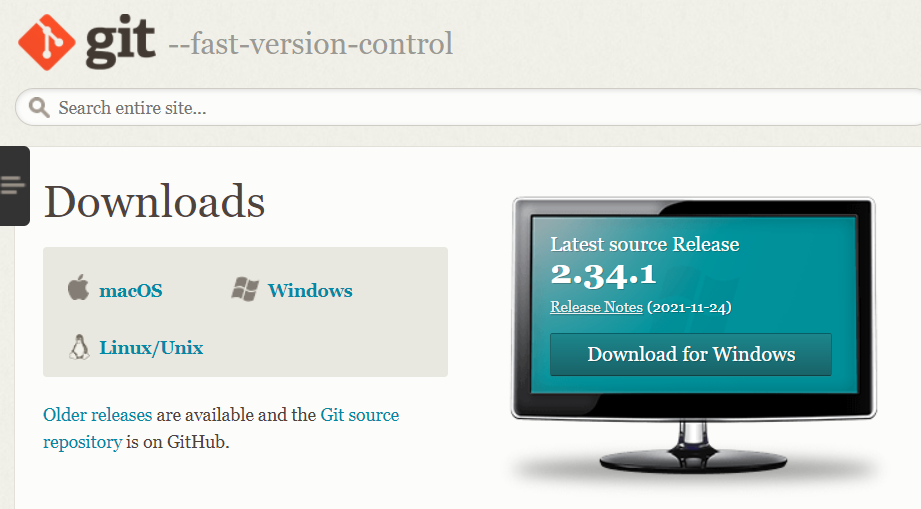
### Вход в приложение

Откроем браузер и введём IP адрес, выключив встроенную защиту браузера. Введём login и password из шага 1.4.

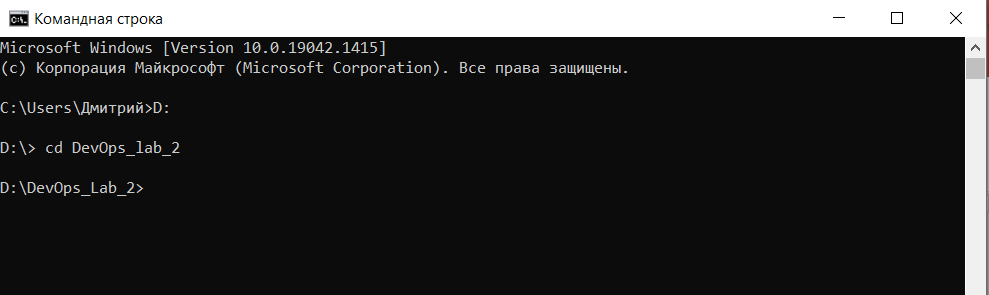


## Загрузка проекта с *https://github.com/olindata/sample-gitlabci-cpp-project на наш сервер Gitlab*.

### Установим 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*

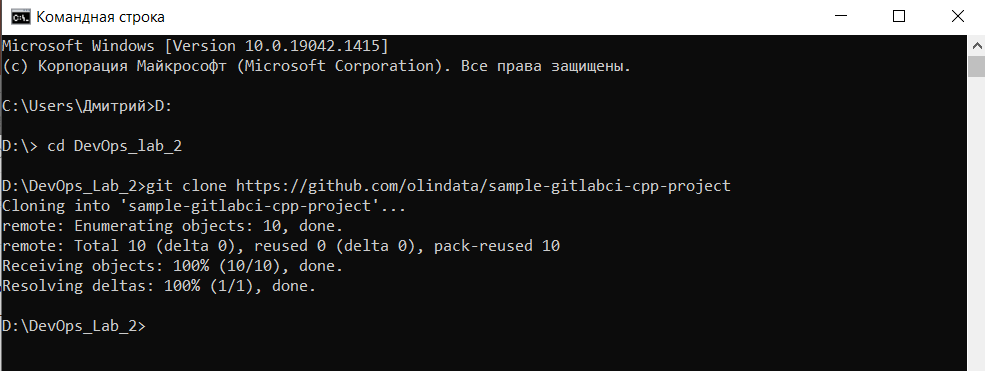


### После установки git-scm откроем терминал и создадим пустую папку, в которую будем клонировать репозиторий с сайта *https://github.com/olindata/sample-gitlabci-cpp-project*



### Клонируем репозиторий в эту папку, при помощи команды:

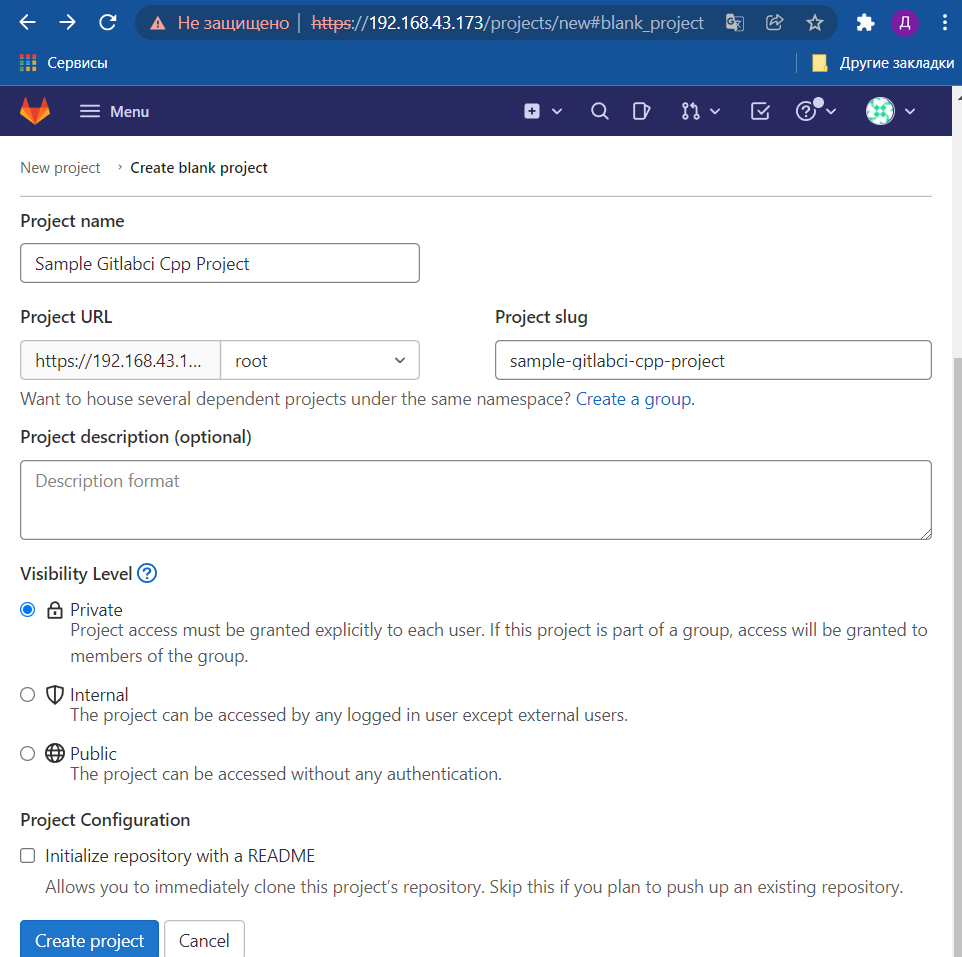
* git clone https://github.com/olindata/sample-gitlabci-cpp-project



### Создадим пустой репозиторий в приложении на сервере.

Параметры репозитория:

* Project slug: **sample-gitlabci-cpp-project**;
* Visibility Level: **Private**;
* Project Configuration: **Initialize repository with a README (снять галочку**).



Скопируем необходимые данные для дальнейшей работы.

Данные для входа при работе с консолью git:

* git config --global user.name "Administrator"
* git config --global user.email admin@example.com

Регистрационный токен для регистрации runner:

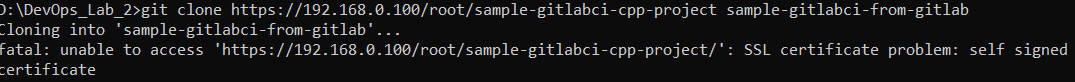
1. Нажмите на проект, над которым вы работаете, на начальном экране
2. Нажмите "Settings" на левой панели навигации
3. Прокрутите вниз до самого низа и нажмите "CI / CD", который вложен в Настройки
4. В разделе "Runners" нажмите кнопку Развернуть
5. Жетон спрятан здесь.

hZ2Bx4BSc1Mx7D8NxzSa

### Клонируем созданный репозиторий к себе на ПК.

В терминале прописываем следующую команду:

* git clone https://192.168.0.100/root/sample-gitlabci-cpp-project sample-gitlabci-from-gitlab.

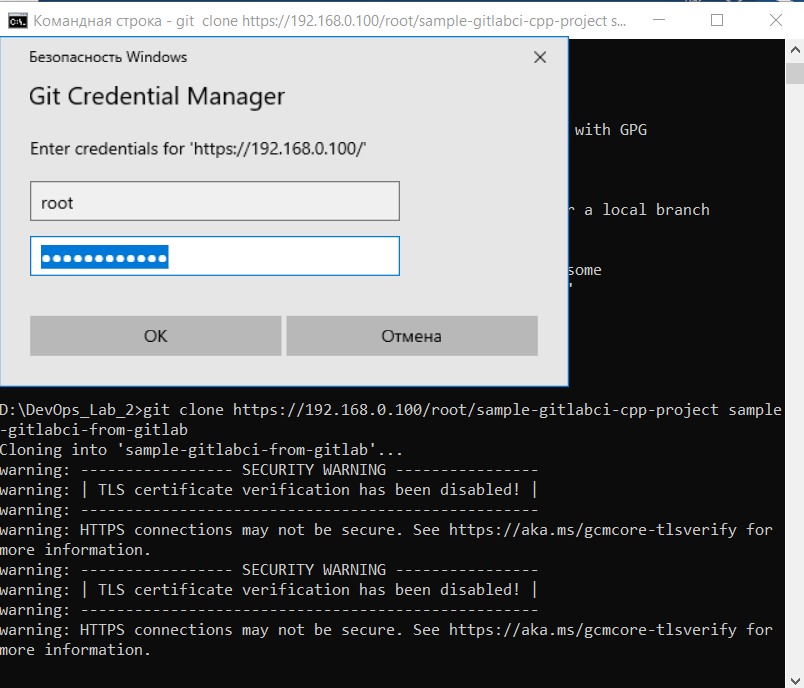


Ошибка “fatal: unable to access 'https://адрес репозитория.git': Peer's Certificate issuer is not recognized” происходит из-за того, что на машине, где выполняется команда, отсутствует доверие к корневому центру сертификации, выдавшему SSL-сертификат, используемый сервером.

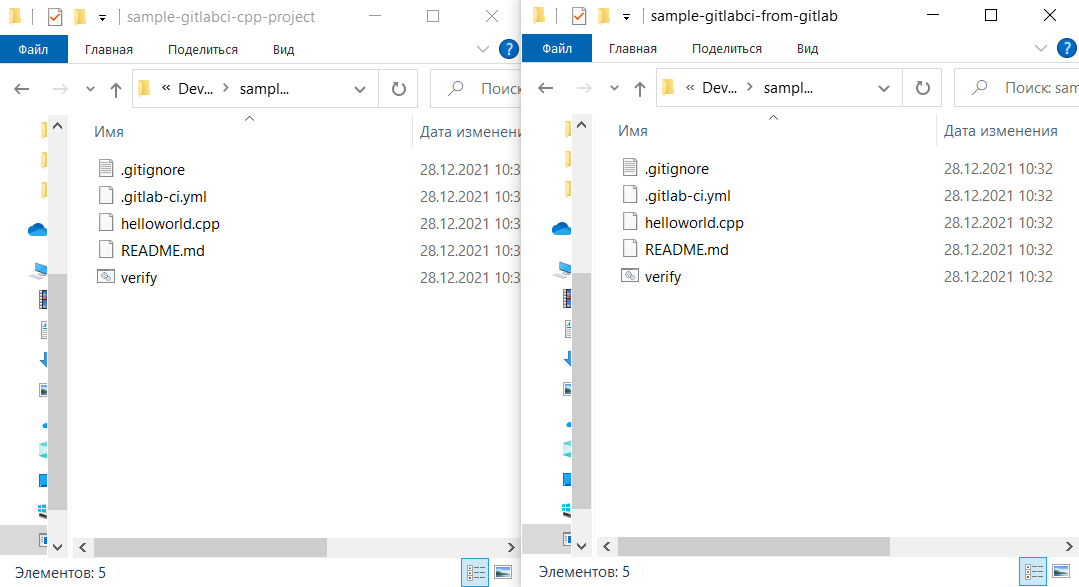
Чтобы временно отключить проверку целостности цепочки сертификации можно использовать параметр **http.sslVerify=false**.. Сделать это можно непосредственно при выполнении команды.

* git config --global http.sslVerify "false"

После вводы команды “git clone https://192.168.0.100/root/sample-gitlabci-cpp-project sample-gitlabci-from-gitlab” необходимо ввести login и password из шага 1.4.

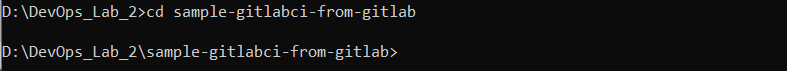
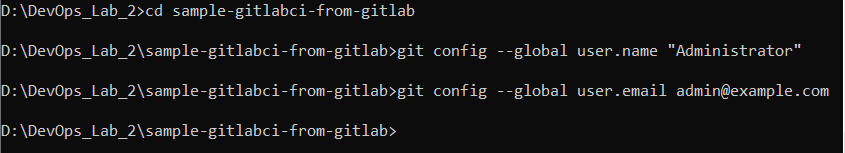


### Для выгрузки проекта на сервер, необходимо скопировать файлы, находящиеся в папке репозитория с GitHub, в папку репозитория GitLab



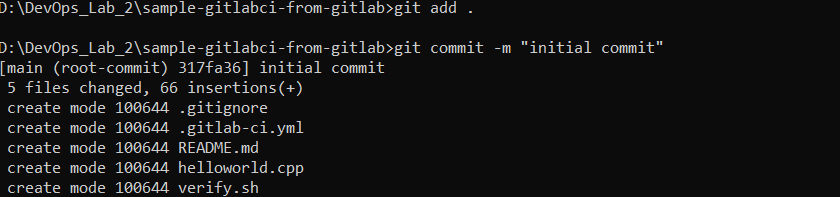
### Перейдем через терминал в папку репозитория GitLab и выполним выгрузку на сервер.

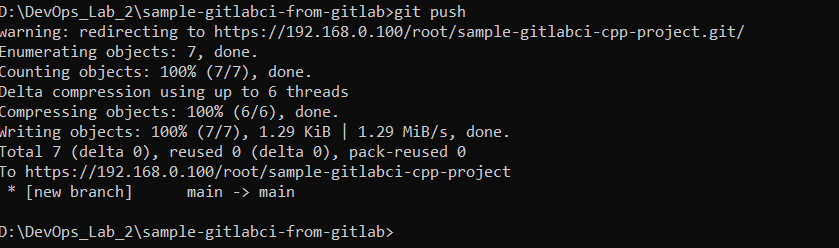
Перед началом выгрузки представимся системе, при помощи следующих команд из шага 2.4

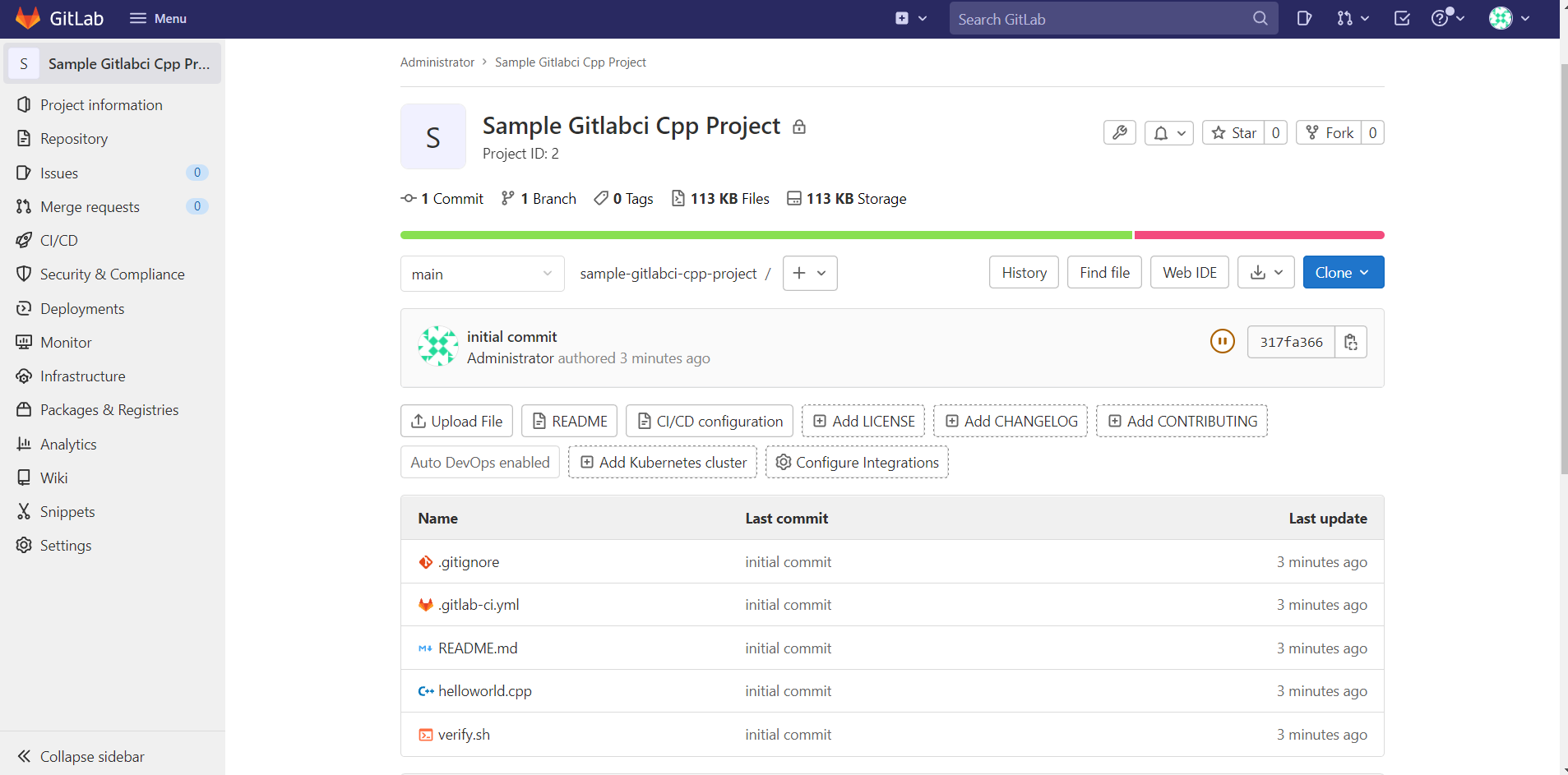
Для выгрузки на сервер выполним следующие команды:

* git add .
* git commit –m “initial commit”
* git push (должно быть подключение к интернету)





### Проверим выгрузку файлов на сервер

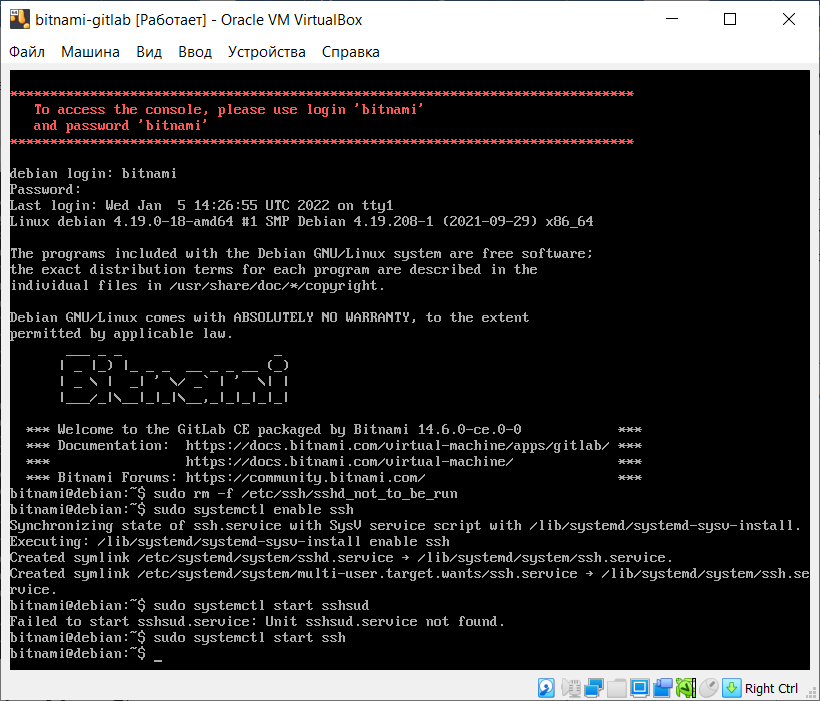


## Разблокируем SSH

По умолчанию на виртуальных машинах Bitnami SSH сервер отключен. Однако нам требуется доступ по SSH для импорта репозиториев кода.

Чтобы включить SSH сервер, выполните в консоли сервера следующие команды:

* sudo rm -f /etc/ssh/sshd\_not\_to\_be\_run
* sudo systemctl enable ssh
* sudo systemctl start ssh



## SSH Public Key

### Отредактируем файл /etc/ssh/sshd\_config командой:

* Sudo nano /etc/ssh/sshd\_config\_