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: «Development Operations»

Тема: **«Git and CI»**

Выполнил: Торжков И.И.

Группа: 6411-100503D

Самара 2022

STEPS

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"
```

7. Reconfigure GitLab for the changes to take effect:

```
$ sudo gitlab-ctl reconfigure
```

8. 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
```

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

```
job:
  script:
      - g++ helloworld.cpp -o helloworld
      - ./verify.sh
```

10. Run Pipeline: **CI/CD > Pipelines > Run pipeline**

PROCEDURE

Download and run Gitlab-Bitnami vmimage.

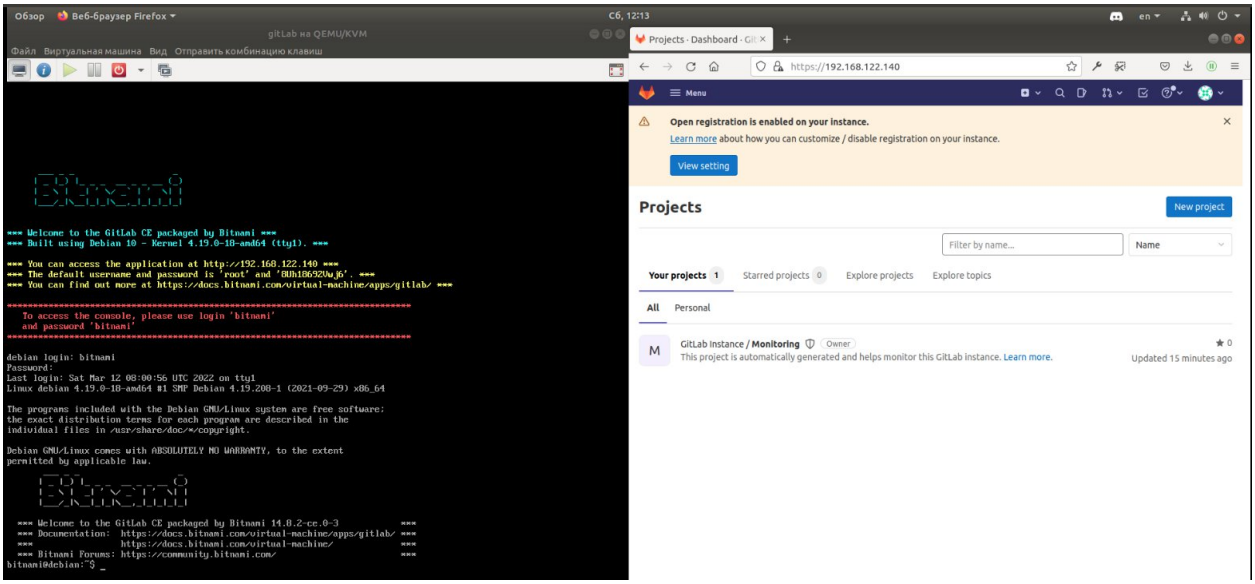

Figure 1

Upload https://github.com/olindata/sample-gitlabci-cpp-project to Gitlab server.

Create new repo.

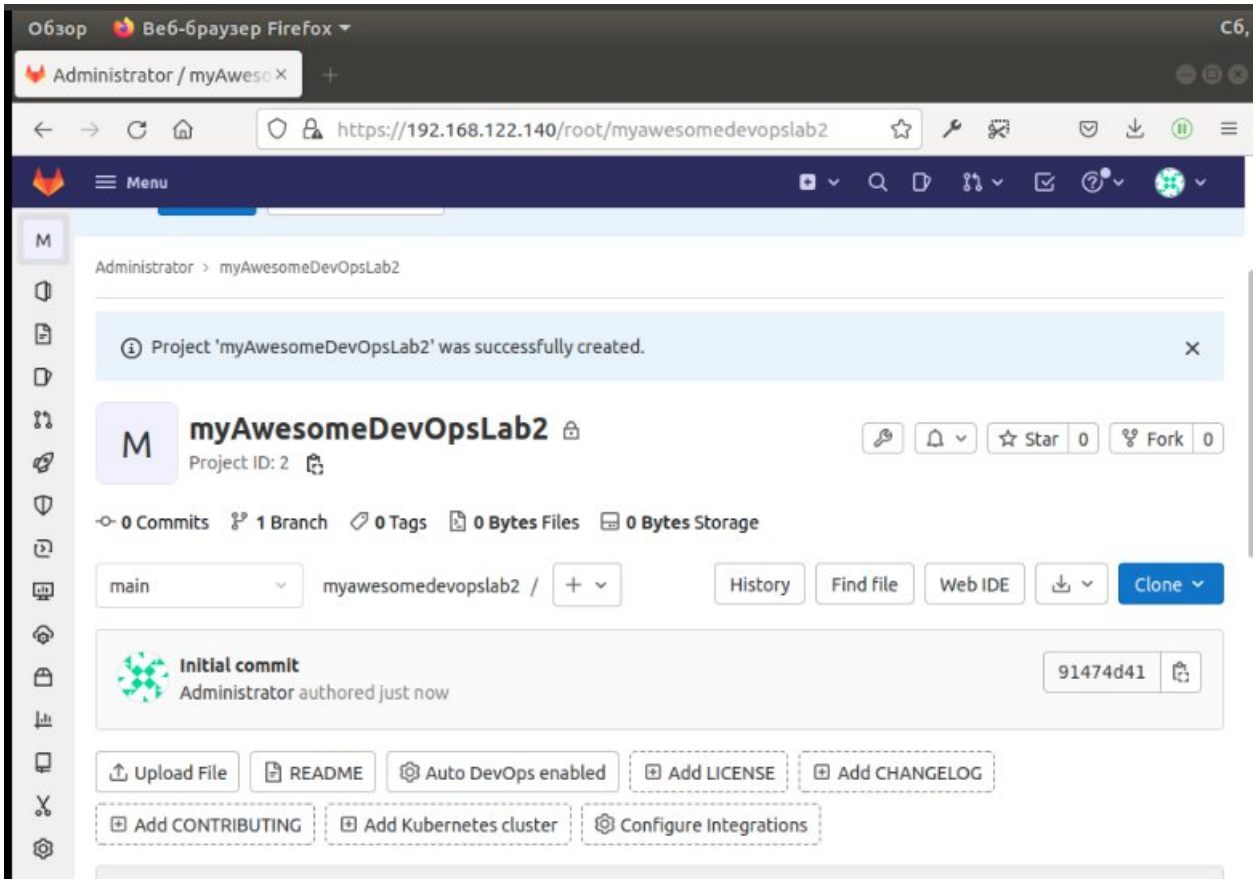

Figure 2

Clone project.

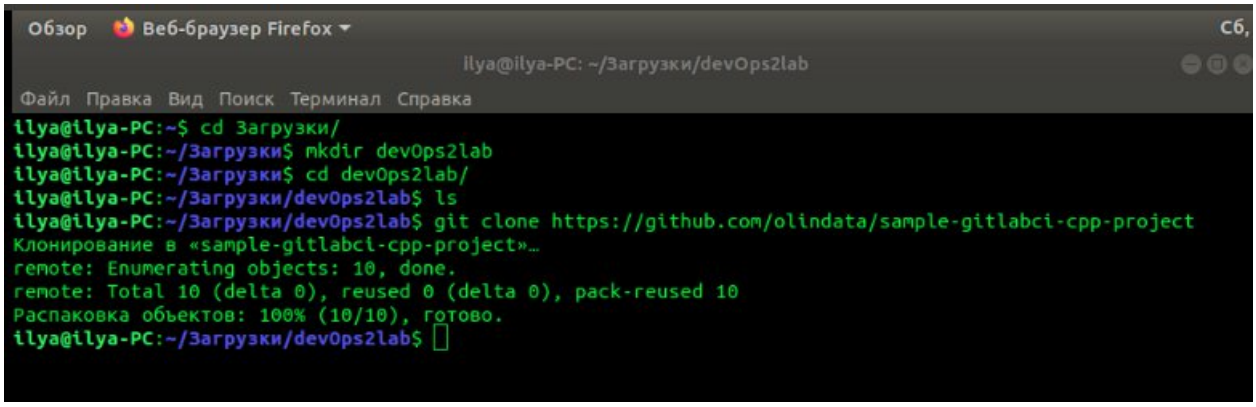

Figure 3

Upload project.

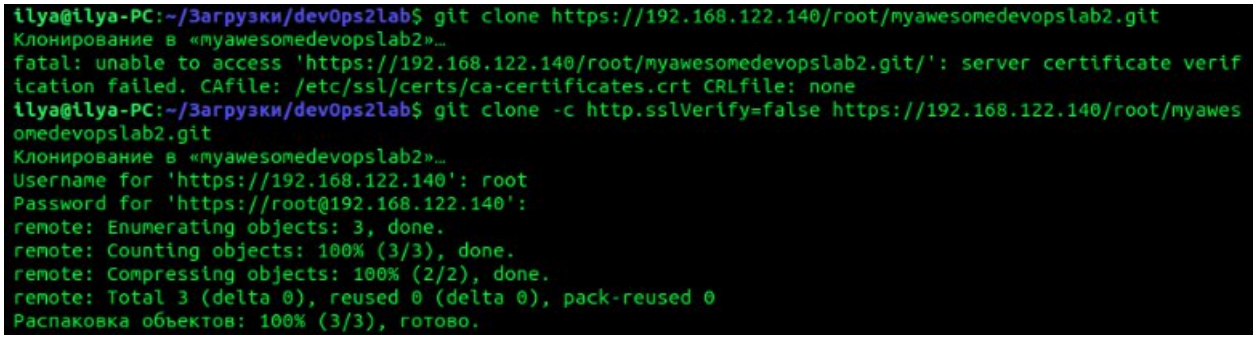

Figure 4

Unblock and configure SSH.

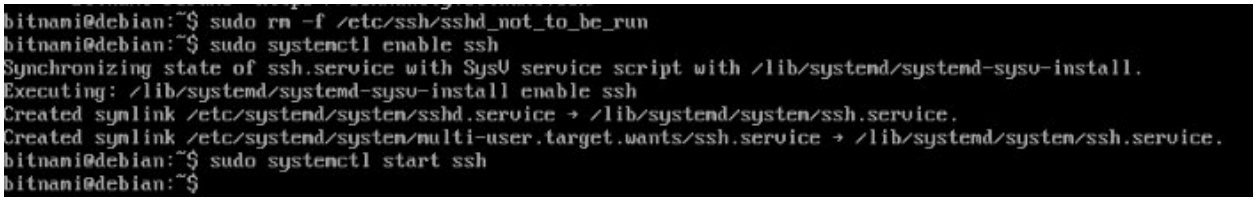

Figure 5

```
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed

#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem       sftp    /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server

ClientAliveInterval 180

Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,chacha20-poly1305@ope

PasswordAuthentication yes

PermitRootLogin no
"/etc/ssh/sshd_config" 129L, 3420C written
bitnami@debian:~$ sudo /etc/init.d/ssh restart
[ ok ] Restarting ssh (via systemctl): ssh.service.
bitnami@debian:~$ _
```

Figure 6

Install GitLab Runner .

Download script from official site.

```
bitnami@debian:~$ curl "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh" | sudo bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5945  100  5945    0     0  16985      0 --:--:-- --:--:-- --:--:-- 16985
Detected operating system as debian/buster.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Running apt-get update... done.
Installing debian-archive-keyring which is needed for installing
apt-transport-https on many Debian systems.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/runner_gitlab-runner.list...done.
Importing packagecloud gpg key... done.
Running apt-get update... done.

The repository is setup! You can now install packages.
bitnami@debian:~$
```

Figure 7

Execute script.

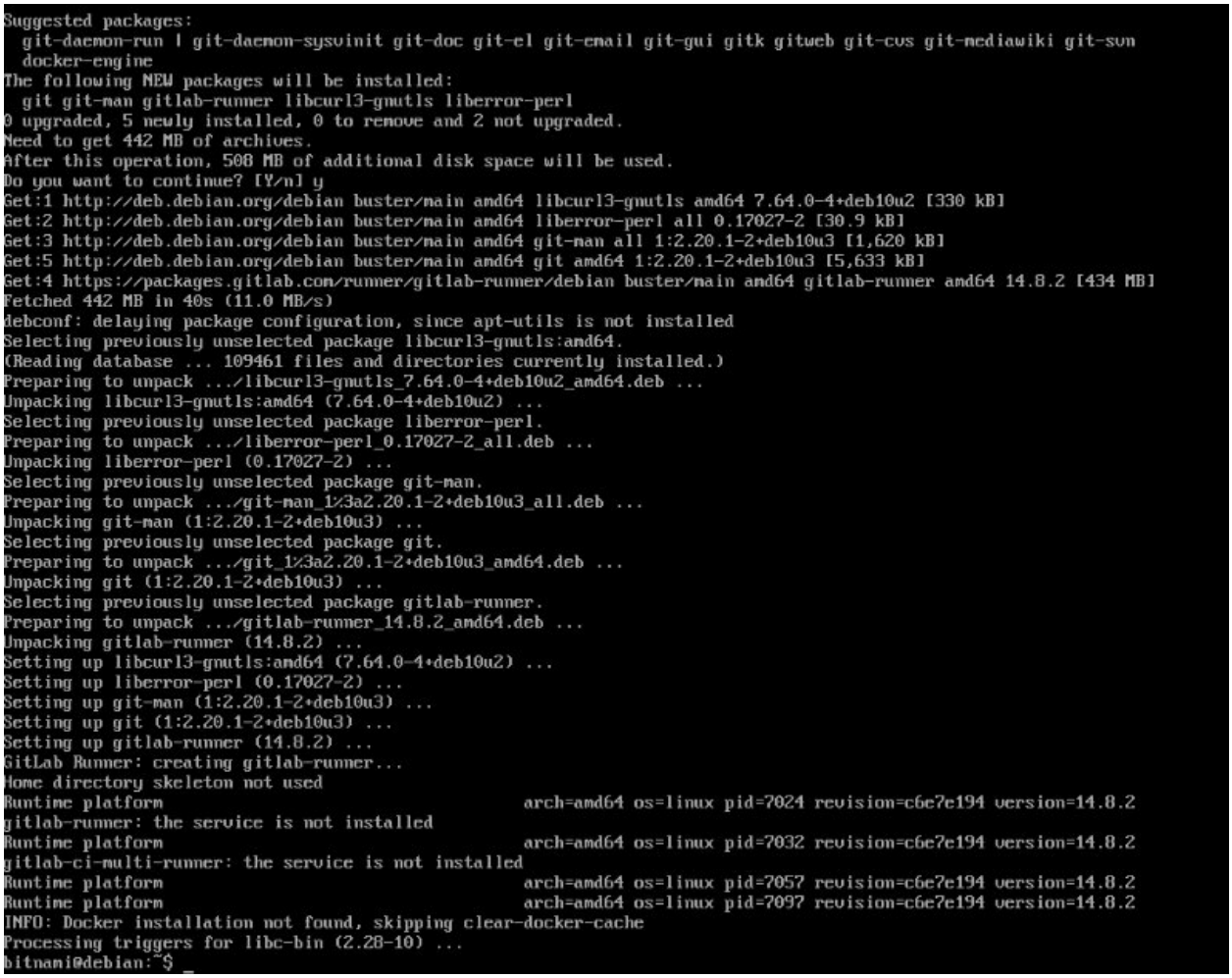

Figure 8

Update /etc/gitlab/gitlab.rb to disable https on gitlab.

external_url 'http://192.168.122 .140 '

nginx['redirect_http_to_https'] = false

nginx['ssl_verify_client'] = "off"

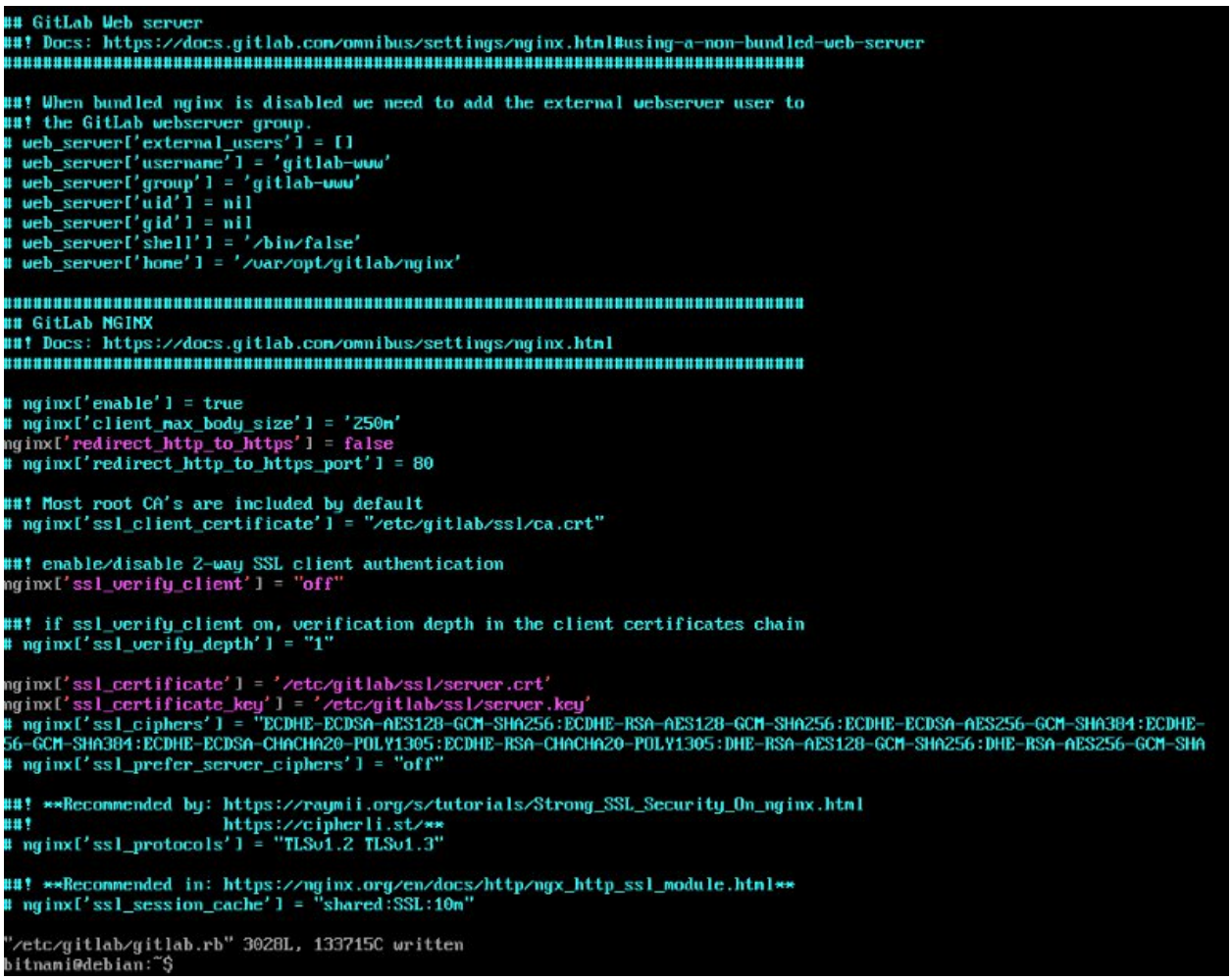
GitLab Web server
##! Docs: https://docs.gitlab.com/omnibus/settings/nginx.html#using-a-non-bundled-web-server
##! When bundled nginx is disabled we need to add the external webserver user to
##! the GitLab webserver group.
web_server['external_users'] = []
web_server['username'] = 'gitlab-www'
web_server['group'] = 'gitlab-www'
web_server['uid'] = nil
web_server['gid'] = nil
web_server['shell'] = '/bin/false'
web_server['home'] = '/var/opt/gitlab/nginx'
GitLab NGINX
##! Docs: https://docs.gitlab.com/omnibus/settings/nginx.html
nginx['enable'] = true
nginx['client_max_body_size'] = '250m'
nginx['redirect_http_to_https'] = false
nginx['redirect_http_to_https_port'] = 80
##! Most root CA's are included by default
nginx['ssl_client_certificate'] = "/etc/gitlab/ssl/ca.crt"
##! enable/disable 2-way SSL client authentication
nginx['ssl_verify_client'] = "off"
##! if ssl_verify_client on, verification depth in the client certificates chain
nginx['ssl_verify_depth'] = "1"
nginx['ssl_certificate'] = '/etc/gitlab/ssl/server.crt'
nginx['ssl_certificate_key'] = '/etc/gitlab/ssl/server.key'
nginx['ssl_ciphers'] = "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-
56-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA
nginx['ssl_prefer_server_ciphers'] = "off"
##! **Recommended by: https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html
##!   https://cipherli.st/**
nginx['ssl_protocols'] = "TLSv1.2 TLSv1.3"
##! **Recommended in: https://nginx.org/en/docs/http/ngx_http_ssl_module.html**
nginx['ssl_session_cache'] = "shared:SSL:10m"
"/etc/gitlab/gitlab.rb" 3028L, 133715C written
bitnami@debian:~$

Figure 9

Reconfigure GitLab.

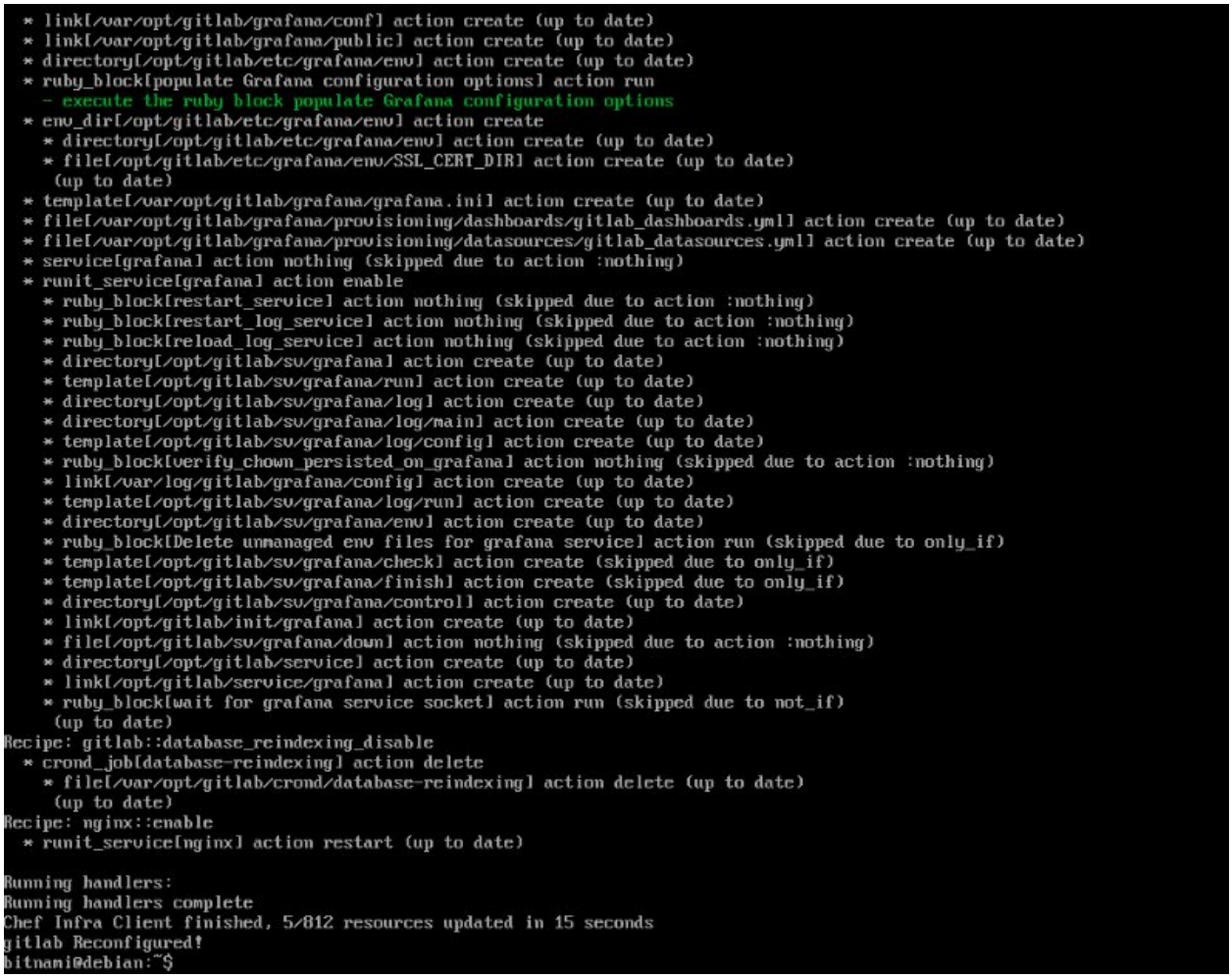

Figure 10

Register runner. Choose shell executor type.

Get token.

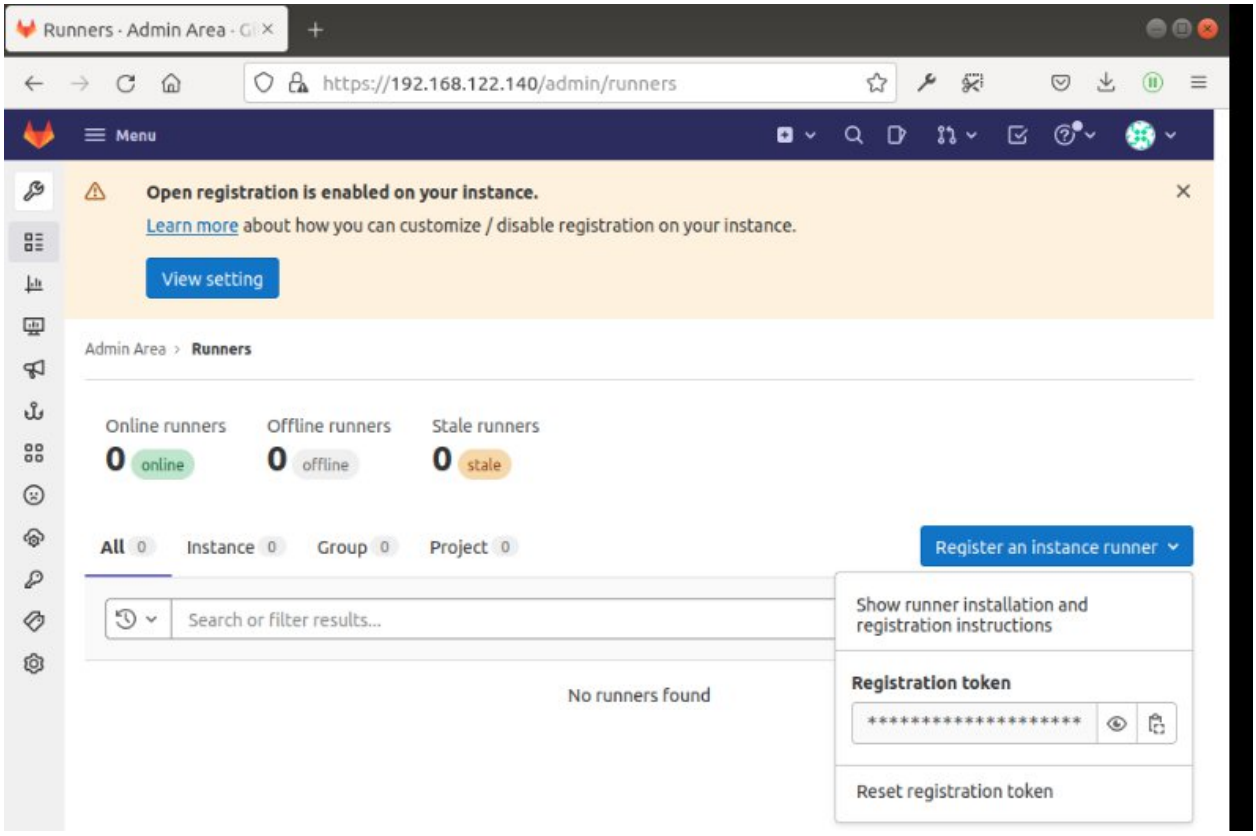

Figure 11

```
bitnami@debian:~$ sudo gitlab-runner register --url http://192.168.122.140:80 --registration-token=FdtJxxyELAsKLJ1xps8C
Runtime platform                                    arch=amd64 os=linux pid=15291 revision=c6e7e194 version=14.8.2
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
[http://192.168.122.140:80]:
Enter the registration token:
[FdtJxxyELAsKLJ1xps8C]:
Enter a description for the runner:
[debian]:
Enter tags for the runner (comma-separated):

Enter optional maintenance note for the runner:

Registering runner... succeeded                     runner=FdtJxxyE
Enter an executor: parallels, shell, kubernetes, docker, docker-ssh, virtualbox, docker+machine, docker-ssh+machine, cus
:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically re
bitnami@debian:~$
```

Figure 12

Edit .gitlab-ci.yml to run runner and run Pipeline.

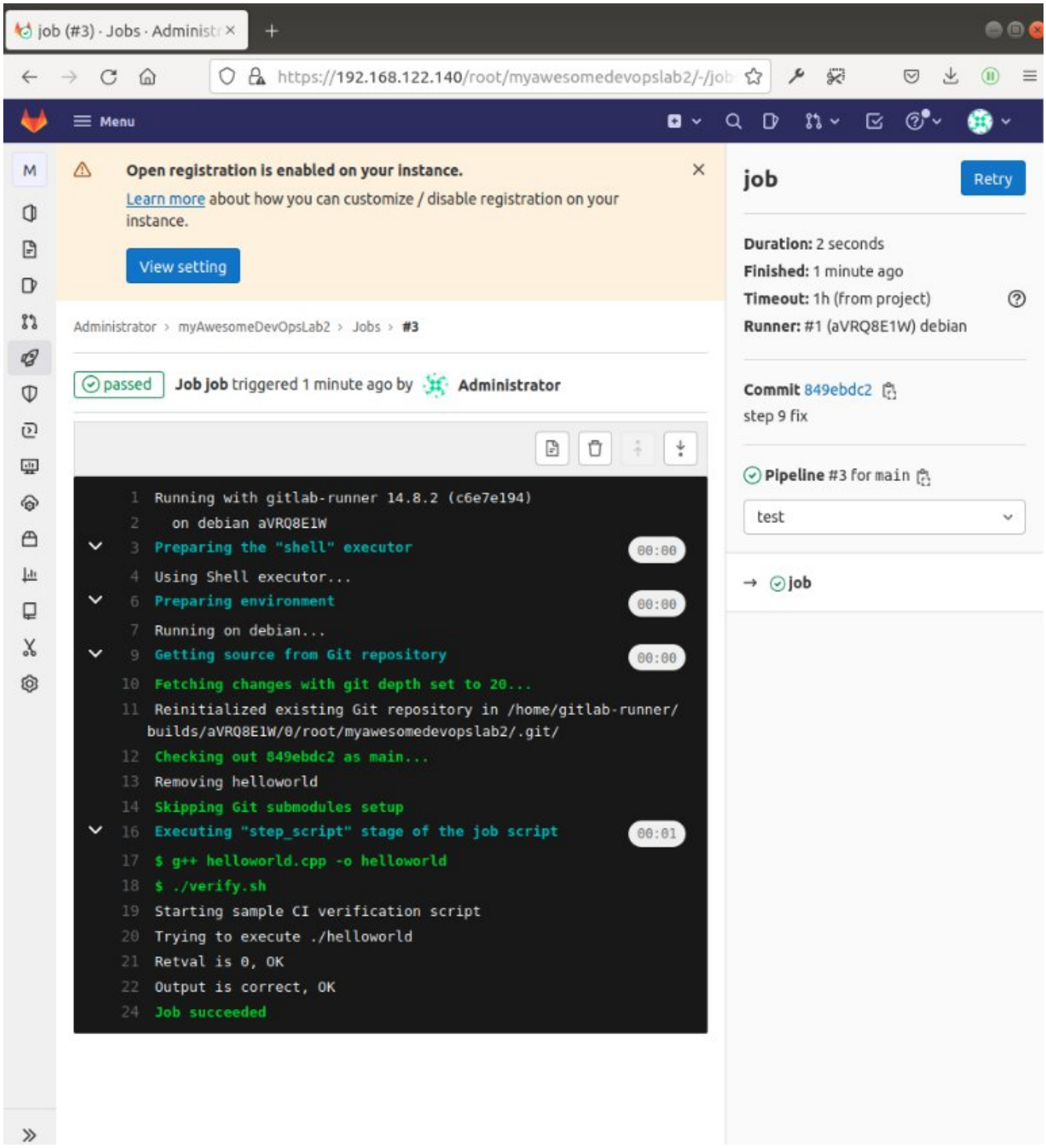

Figure 13

CONCLUSION

During the execution of the work, GitLab server was created, automatic software compilation and verification was configured, basic utilities for working with repositories, runners and pipelines were studied.