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: Харитонов Н.В.

Группа: 6133-010402D

Самара 2021

TASK

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
```

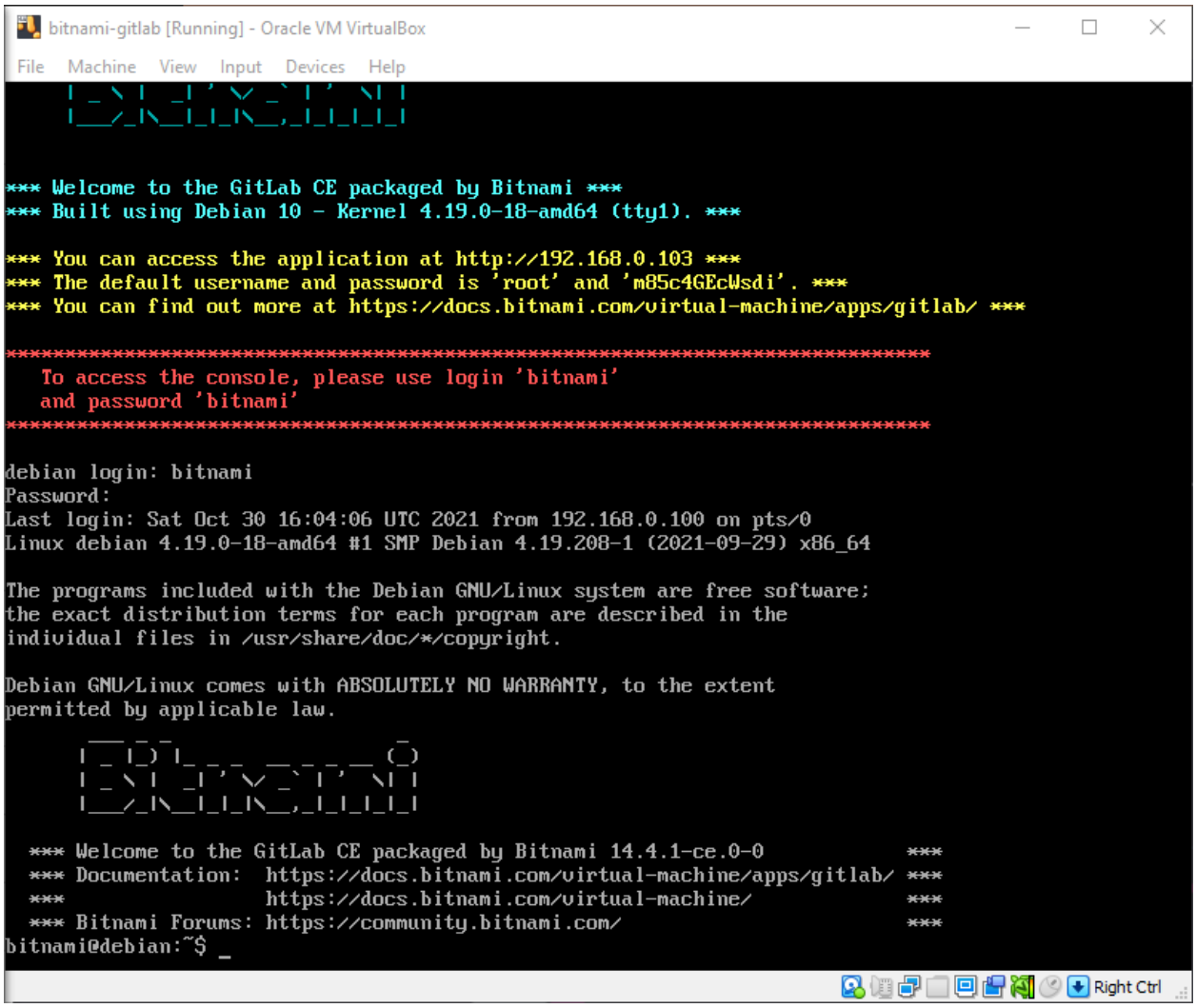
9. Edit .gitlab-ci.yml to run runner in shell mode (without Docker)

```
job:
script:
g++ helloworld.cpp -o helloworld
```

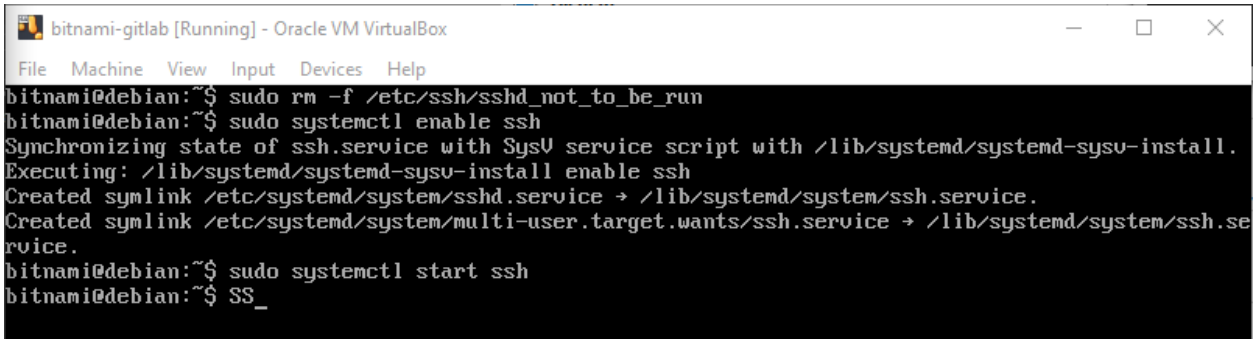
10. Run Pipeline: CI/CD > Pipelines > Run pipeline

PROCEDURE

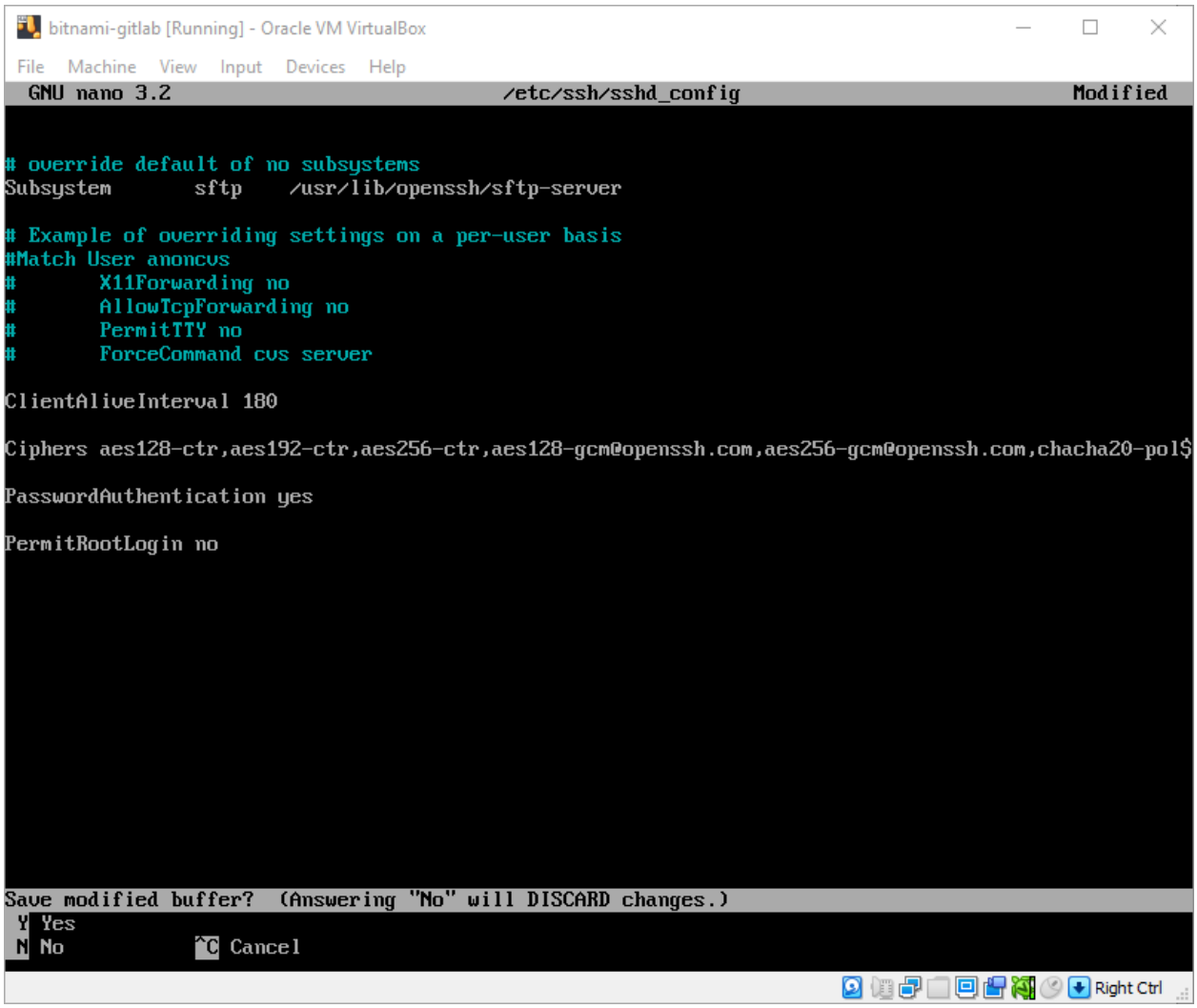
Download and open the Gitlab-Bitnami VM:

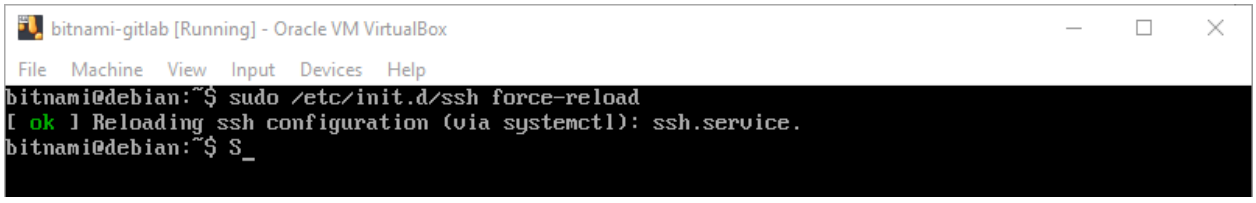


Enable SSH on that VM:

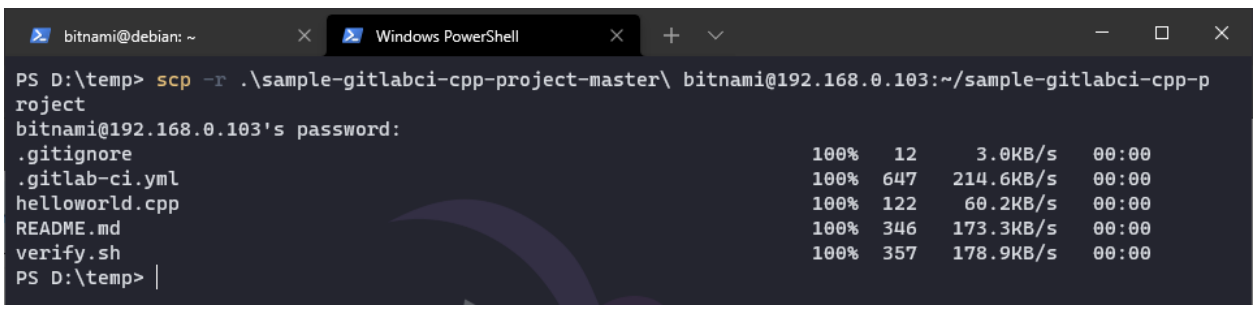


Edit the SSH configuration file to support password authentication:

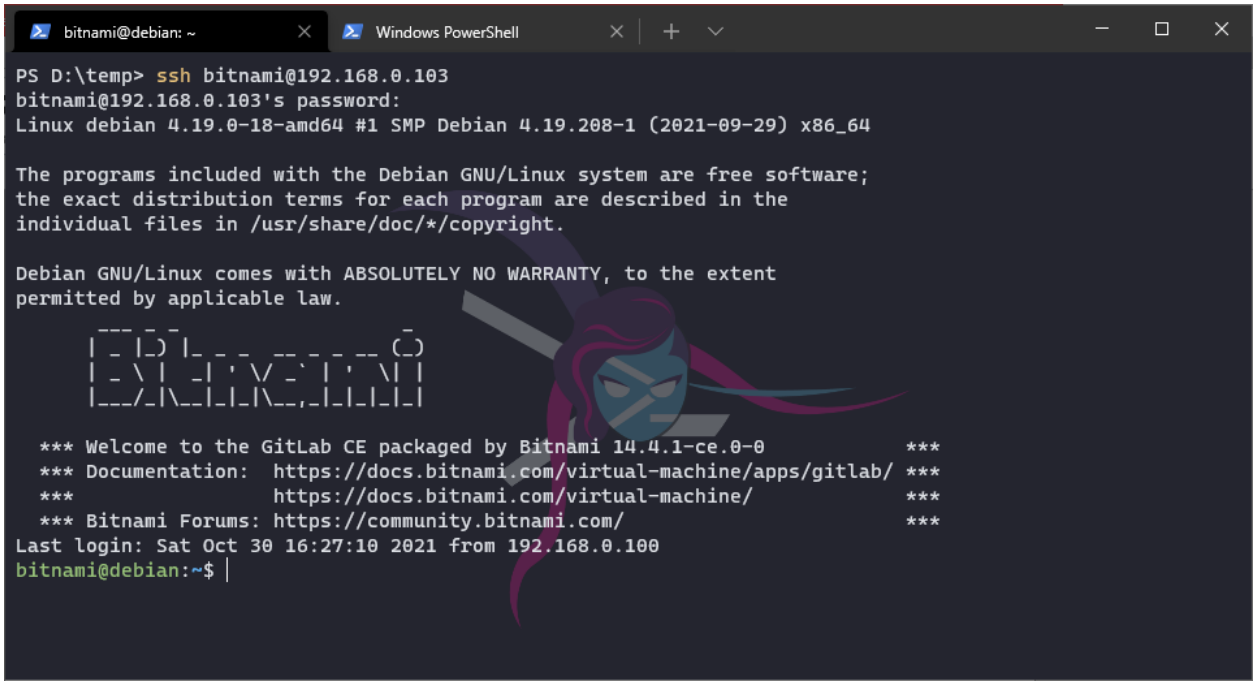




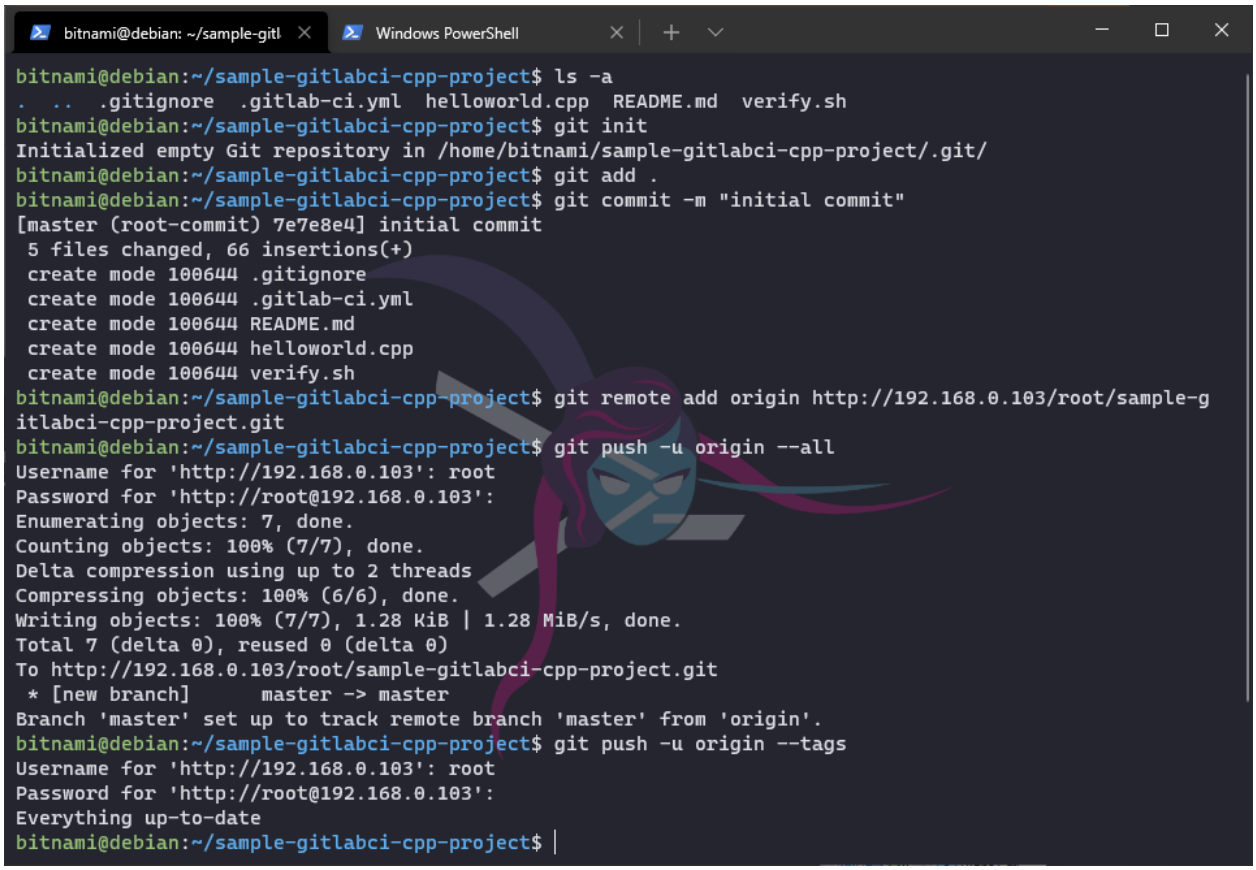
Clone the project to the host machine and then copy it to the VM via SSH:



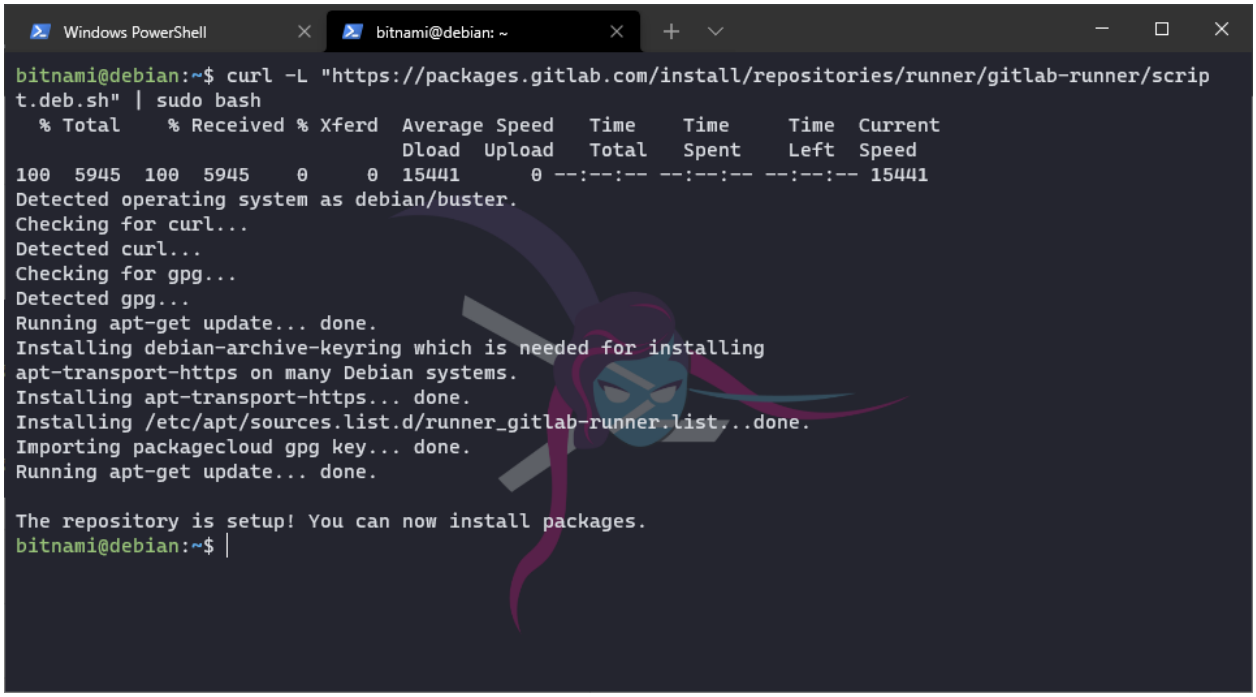
Connect to the VM via SSH:

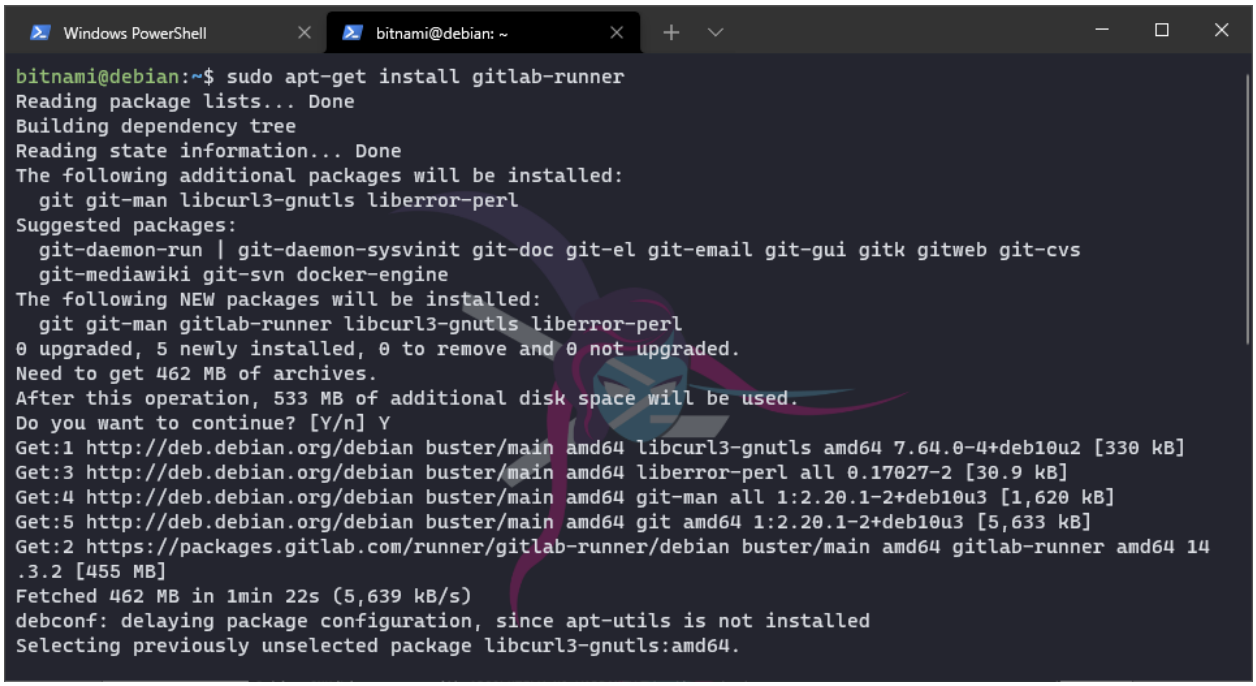


Once connected, go to the project directory and initialize the Git repository in it. Then push it to the GitLab repository.

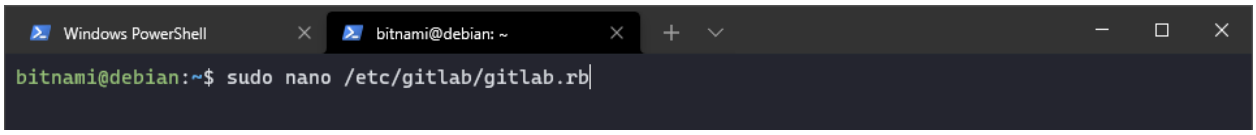


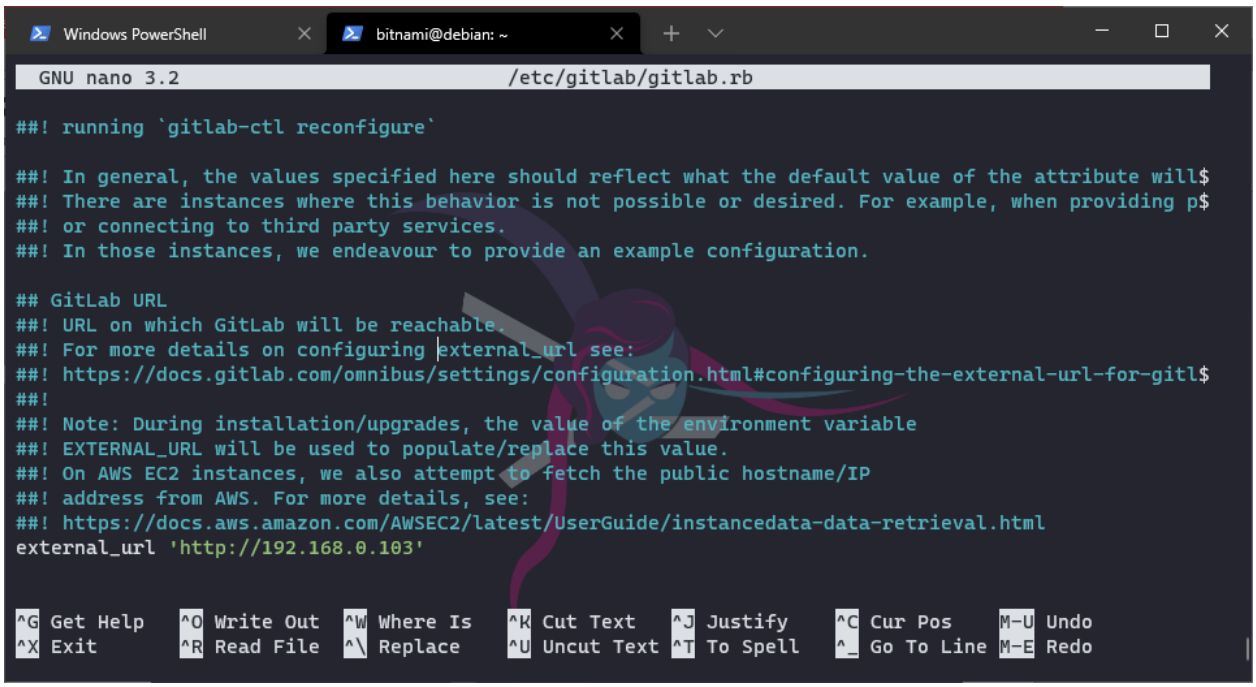
Install GitLab Runner:

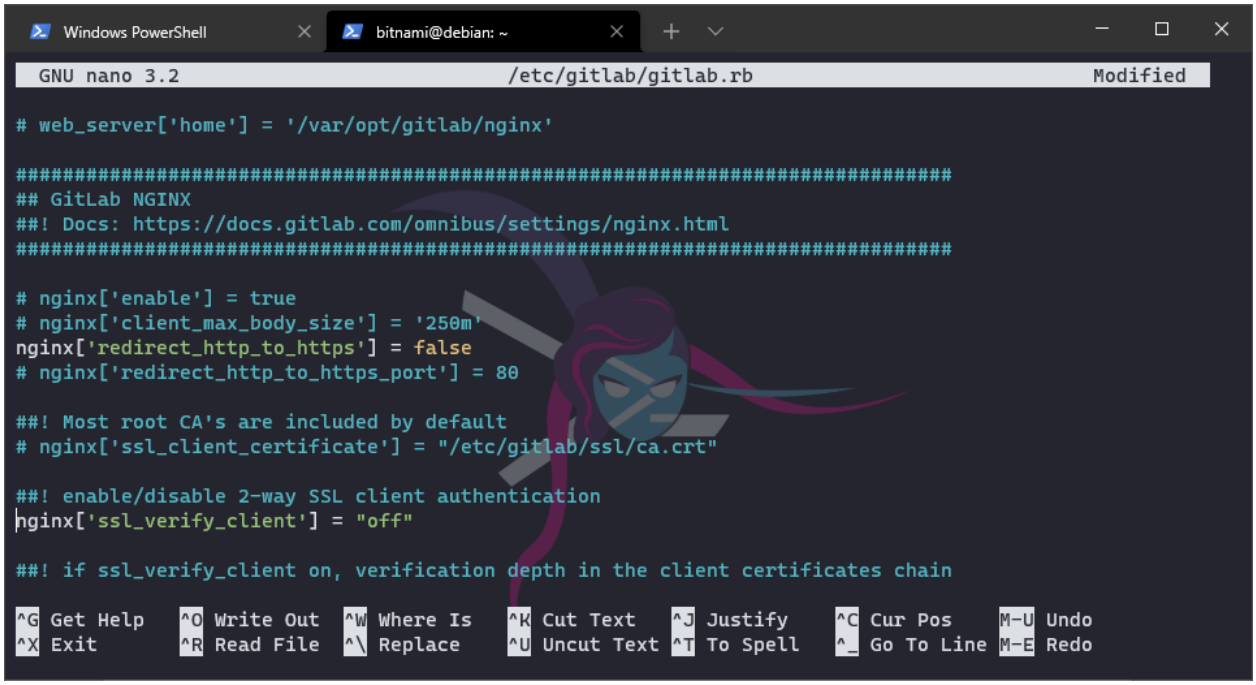




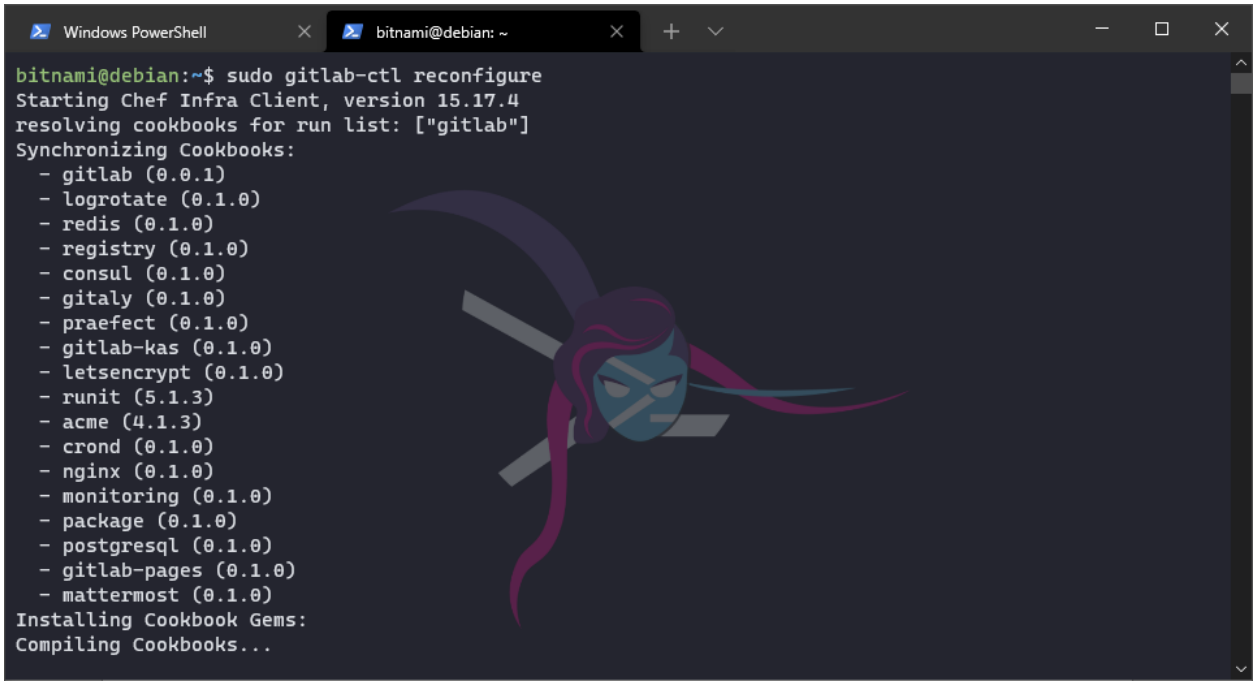
Update the `/etc/gitlab/gitlab.rb` file to disable HTTPS on GitLab:



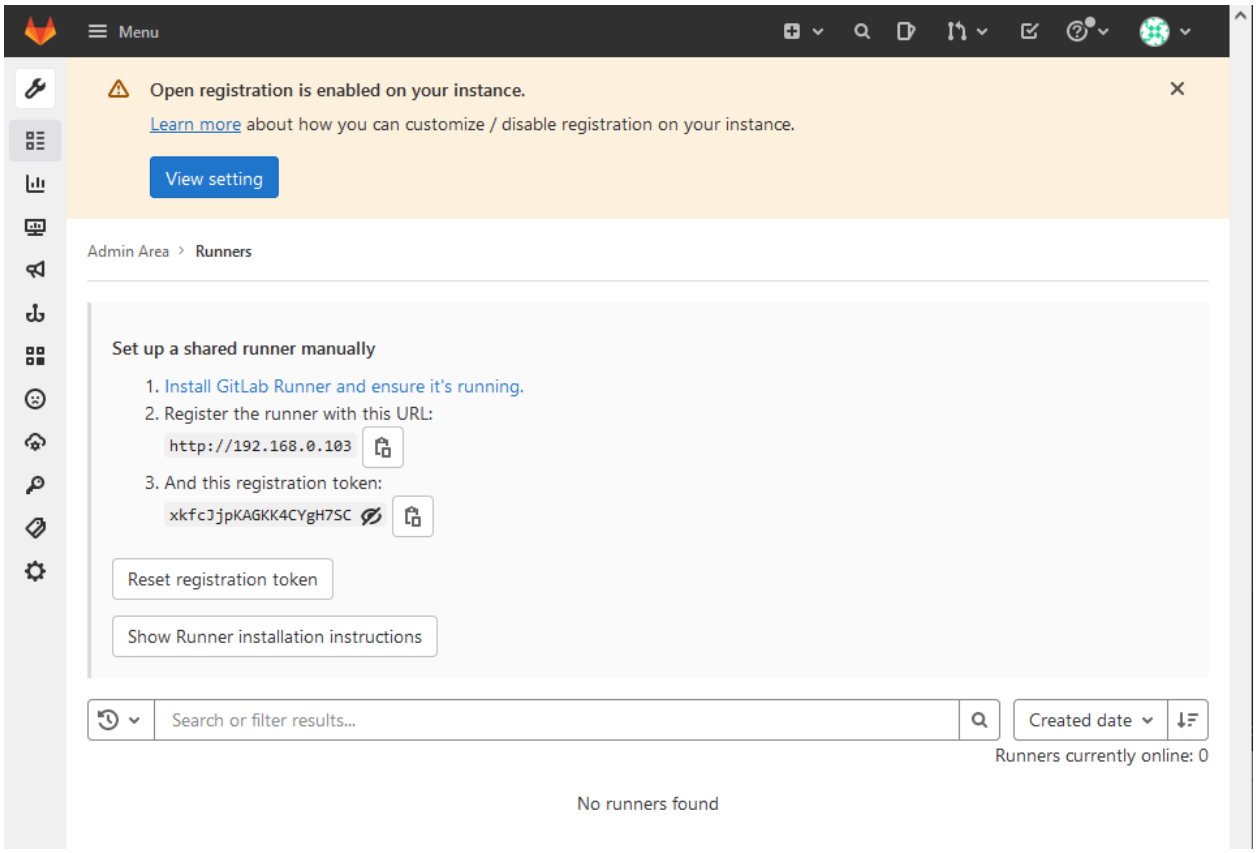

Windows PowerShell
bitnami@debian: ~
GNU nano 3.2 /etc/gitlab/gitlab.rb
##! running `gitlab-ctl reconfigure`
##! In general, the values specified here should reflect what the default value of the attribute will$
##! There are instances where this behavior is not possible or desired. For example, when providing p$
##! or connecting to third party services.
##! In those instances, we endeavour to provide an example configuration.
GitLab URL
##! URL on which GitLab will be reachable.
##! For more details on configuring external_url see:
##! https://docs.gitlab.com/omnibus/settings/configuration.html#configuring-the-external-url-for-gitl$
##!
##! Note: During installation/upgrades, the value of the environment variable
##! EXTERNAL_URL will be used to populate/replace this value.
##! On AWS EC2 instances, we also attempt to fetch the public hostname/IP
##! address from AWS. For more details, see:
##! https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html
external_url 'http://192.168.0.103'
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo


Windows PowerShell
bitnami@debian: ~
GNU nano 3.2 /etc/gitlab/gitlab.rb Modified
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
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo

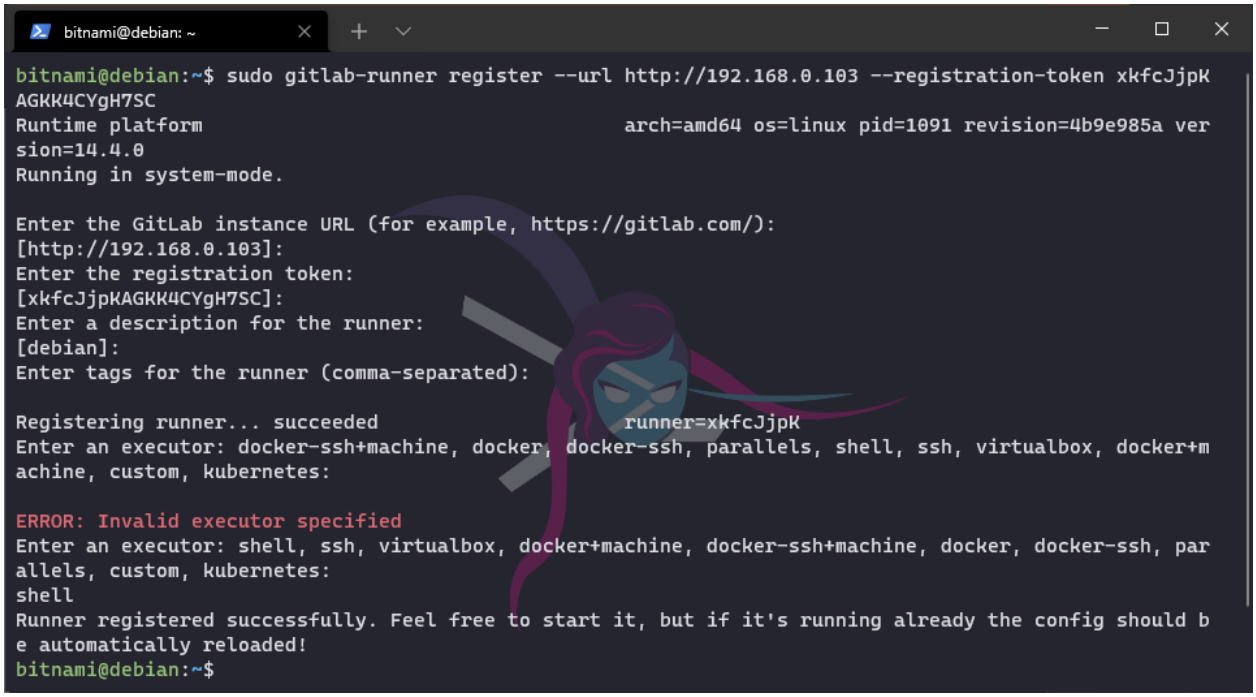
Reconfigure GitLab for the changes to take effect:



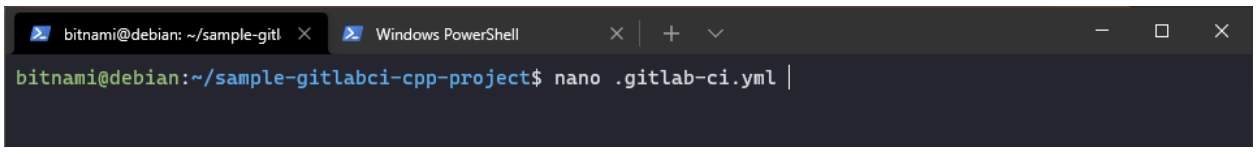
Open the GitLab server to get the runner registration token:

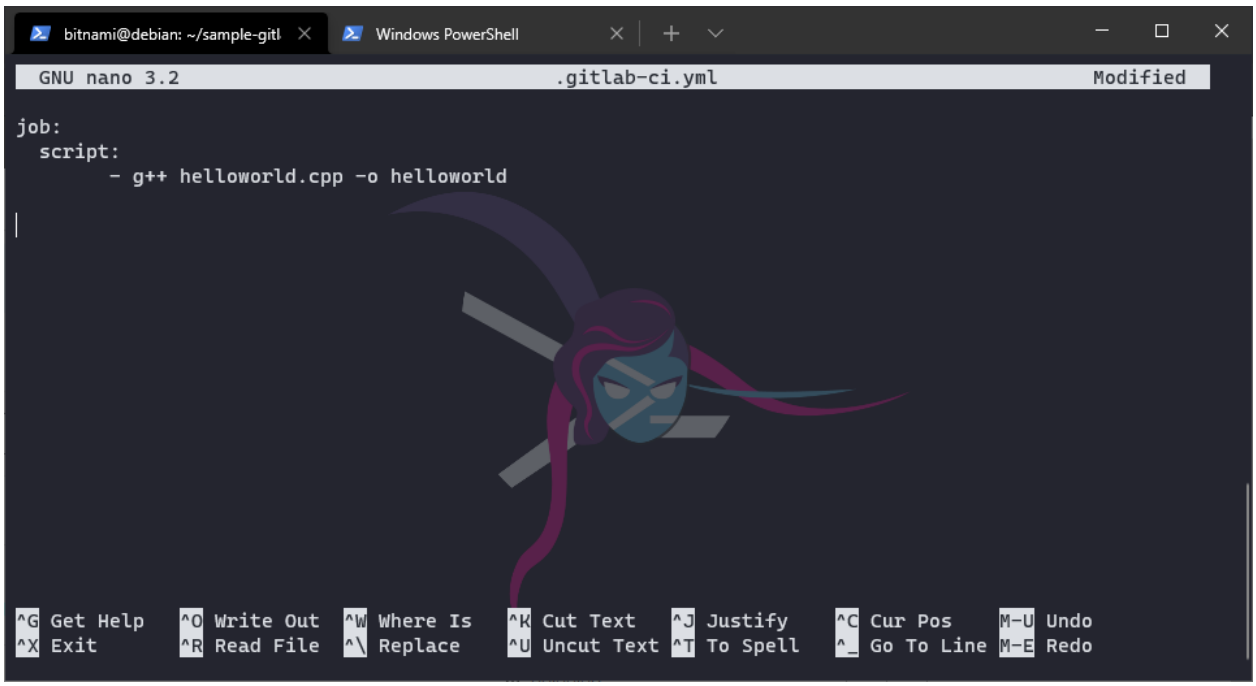


Register a runner:

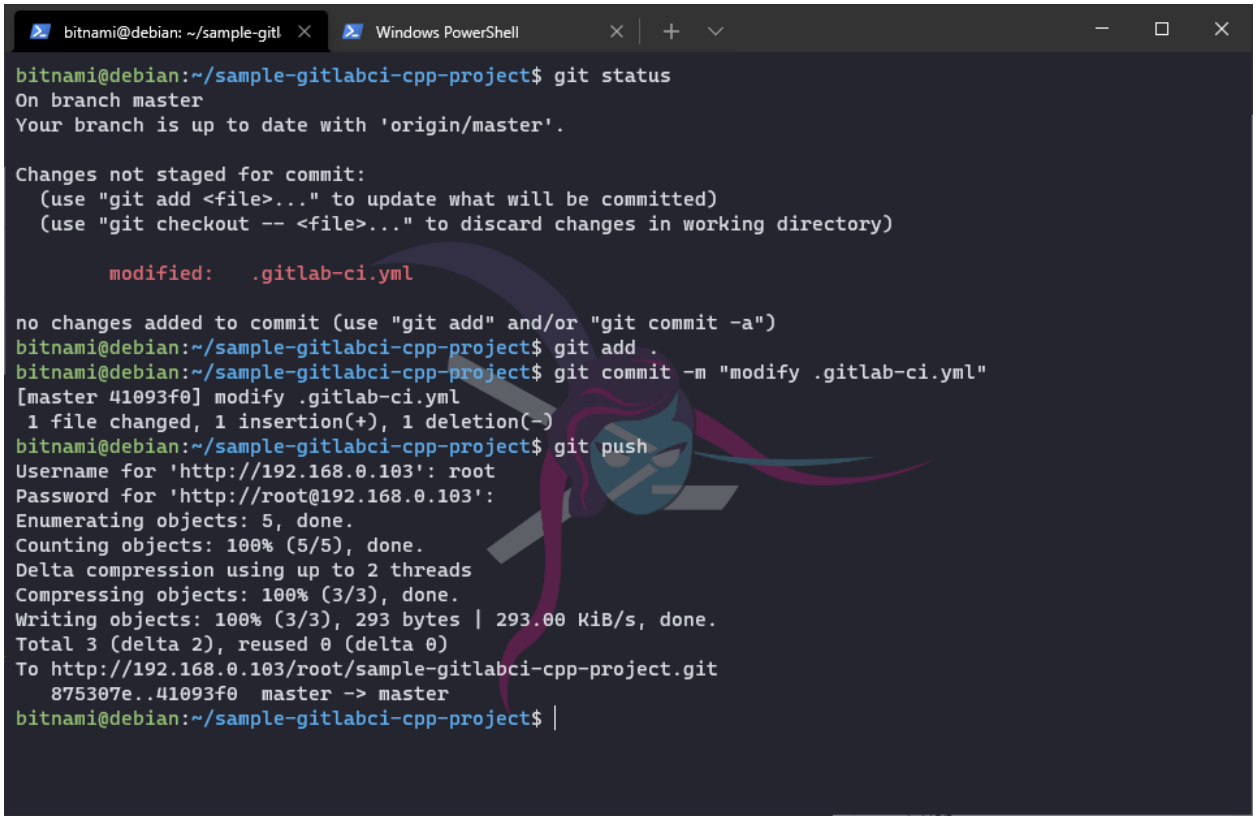


In the project directory, edit `.gitlab-ci.yml` to run runner in shell mode:

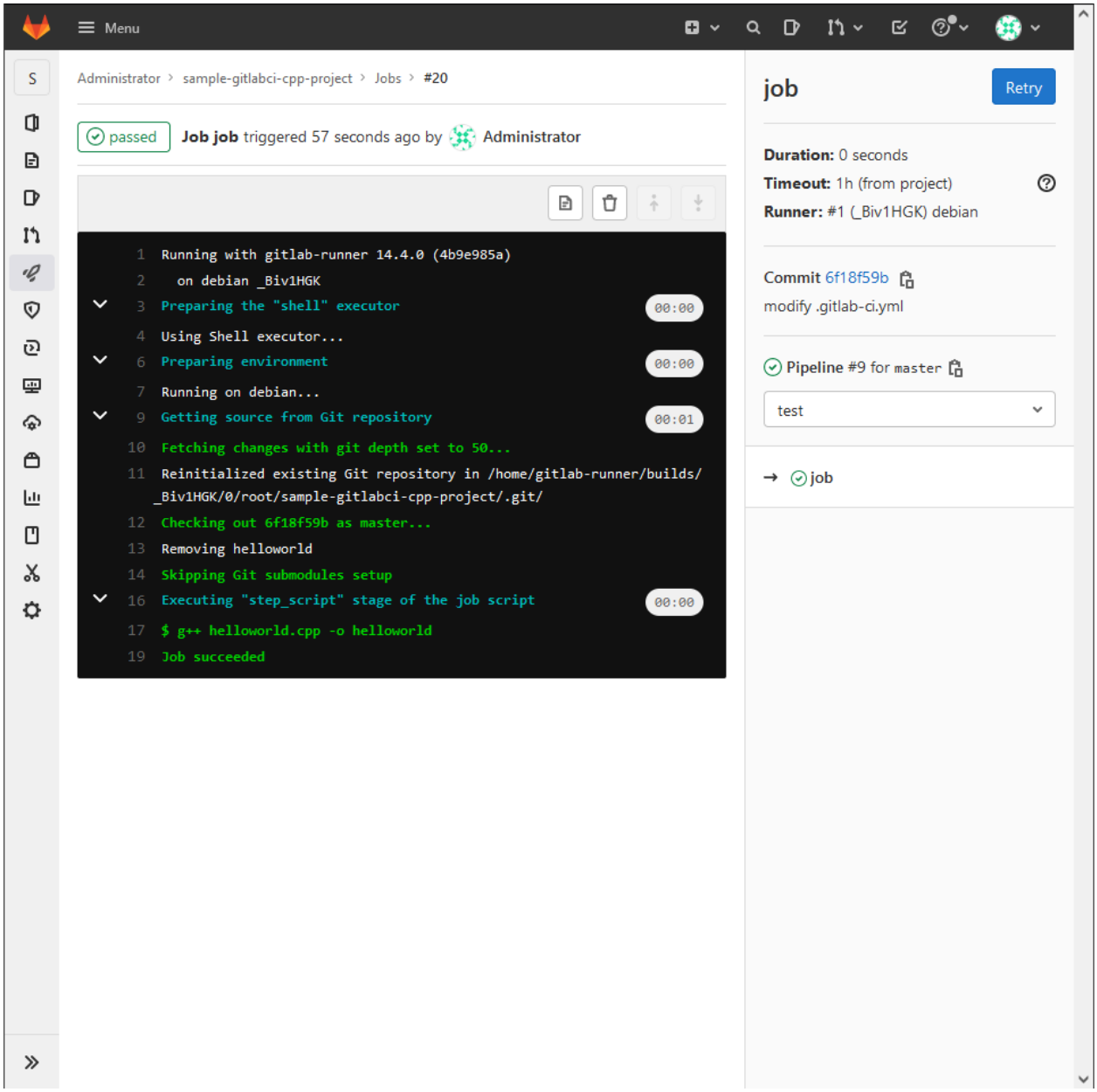




Commit and push the changes to the server:



Check if the pipeline runs successfully or not:



The pipeline runs successfully.

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.