


## SPRAWOZDANIE NR 2

Nazwa ćwiczenia	Konfiguracja gitlab przez docker-compose		 <b>POLITECHNIKA BYDGOSKA</b> Wydział Telekomunikacji, Informatyki i Elektrotechniki
Przedmiot	Narzędzia Programistyczne		
Student grupa	Maciej Kwiatkowski,Dorian Zasada,Jakub Weiss,Julian Frishke		
Data ćwiczeń			Data oddania sprawozdania

Gitlab jest to rozwiązanie do przechowywania repozytorium Git w chmurze. Pracując nad nowym oprogramowaniem niezbędne jest korzystanie kontroli wersji dlatego Gitlab jest cały czas rozwijany.

Instalacja GitLab poprzez Docker wymaga zainstalowania wspomnianego narzędzia wykonujemy to komendą

Aktualizujemy pakiety i instalujemy dockera

```
maciej@maciej:~$ sudo apt-get update
[sudo] password for maciej:
Stary:1 http://pl.archive.ubuntu.com/ubuntu jammy InRelease
Pobieranie:2 http://pl.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Stary:3 http://pl.archive.ubuntu.com/ubuntu jammy-backports InRelease
Pobieranie:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Pobieranie:5 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1 608 kB]
Pobieranie:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1 388 kB]
Pobieranie:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [243 kB]
Pobieranie:8 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1 762 kB]
Pobieranie:9 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [299 kB]
Pobieranie:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [848 kB]
Pobieranie:11 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [303 kB]
Pobieranie:12 http://pl.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1 826 kB]
Pobieranie:13 http://pl.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [310 kB]
Pobieranie:14 http://pl.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1 070 kB]
Pobrano 9 885 kB w 6s (1 554 kB/s)
Czytanie list pakietów... Gotowe
maciej@maciej:~$
```

```
maciej@maciej:~$ sudo apt-get install docker.io
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Sugerowane pakiety:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
Zostaną zainstalowane następujące NOWE pakiety:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 aktualizowanych, 8 nowo instalowanych, 0 usuwanych i 4 nieaktualizowanych.
Konieczne pobranie 69,8 MB archiwów.
Po tej operacji zostanie dodatkowo użyte 267 MB miejsca na dysku.
Kontynuować? [T/n] T
Pobieranie:1 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63,6 kB]
Pobieranie:2 http://pl.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34,4 kB]
Pobieranie:3 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.7-0ubuntu1~22.04.2 [4 267 kB]
Pobieranie:4 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.2-0ubuntu1~22.04.1 [36,0 MB]
Ign:4 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.2-0ubuntu1~22.04.1
Pobieranie:5 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dns-root-data all 2023112702~ubuntu0.22.04.1 [5 136 B]
Pobieranie:6 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main amd64 dnsmasq-base amd64 2.90-0ubuntu0.22.04.1 [374 kB]
Pobieranie:7 http://pl.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 24.0.5-0ubuntu1~22.04.1 [28,9 MB]
Pobieranie:8 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35,2 kB]
Pobieranie:4 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.7.2-0ubuntu1~22.04.1 [36,0 MB]
Pobrano 56,6 MB w 45s (1 256 kB/s)
Prekonfiguracja pakietów...
```

Przystępujemy do instalacji gitlaba w kilku krokach

## 1)Konfigurujemy SSH

```
maciej@maciej:~$ sudo apt-get install -y curl openssh-server ca-certificates tzdata perl
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates zaznaczony jako zainstalowany ręcznie.
curl is already the newest version (7.81.0-1ubuntu1.16).
curl zaznaczony jako zainstalowany ręcznie.
openssh-server is already the newest version (1:8.9p1-3ubuntu0.7).
perl is already the newest version (5.34.0-3ubuntu1.3).
perl zaznaczony jako zainstalowany ręcznie.
tzdata is already the newest version (2024a-0ubuntu0.22.04).
tzdata zaznaczony jako zainstalowany ręcznie.
```

Instalujemy pakiet pocztowy postfix

```
maciej@maciej:~$ sudo apt-get install -y postfix
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
The following additional packages will be installed:
  ssl-cert
Sugerowane pakiety:
  procmail postfix-mysql postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb postfix-sqlite sasl2-bin | dovecot-common resolvconf
  postfix-cdb mail-reader postfix-mta-sts-resolver postfix-doc
Zostaną zainstalowane następujące NOWE pakiety:
  postfix ssl-cert
0 aktualizowanych, 2 nowo instalowanych, 0 usuwanych i 4 nieaktualizowanych.
Konieczne pobranie 1 265 kB archiwów.
Po tej operacji zostanie dodatkowo użyte 4 248 kB miejsca na dysku.
Pobieranie:1 http://pl.archive.ubuntu.com/ubuntu/jammy/main/amd64/ssl-cert/all/1.1.2 [17,4 kB]
Pobieranie:2 http://pl.archive.ubuntu.com/ubuntu/jammy-updates/main/amd64/postfix/amd64/3.6.4-1ubuntu1.3 [1 248 kB]
Ign:2 http://pl.archive.ubuntu.com/ubuntu/jammy-updates/main/amd64/postfix/amd64/3.6.4-1ubuntu1.3
Pobieranie:2 http://pl.archive.ubuntu.com/ubuntu/jammy-updates/main/amd64/postfix/amd64/3.6.4-1ubuntu1.3 [1 248 kB]
Ign:2 http://pl.archive.ubuntu.com/ubuntu/jammy-updates/main/amd64/postfix/amd64/3.6.4-1ubuntu1.3
Pobieranie:2 http://pl.archive.ubuntu.com/ubuntu/jammy-updates/main/amd64/postfix/amd64/3.6.4-1ubuntu1.3 [1 248 kB]
Pobrano 877 kB w 19s (47,2 kB/s)
Prekonfiguracja pakietów ...
Wybieranie wcześniej niewybranego pakietu ssl-cert.
(Odczytywanie bazy danych ... 74917 plików i katalogów obecnie zainstalowanych.)
Przygotowywanie do rozpakowania pakietu .../ssl-cert_1.1.2_all.deb ...
Rozpakowywanie pakietu ssl-cert (1.1.2) ...
Wybieranie wcześniej niewybranego pakietu postfix.
Przygotowywanie do rozpakowania pakietu .../postfix_3.6.4-1ubuntu1.3_amd64.deb ...
Rozpakowywanie pakietu postfix (3.6.4-1ubuntu1.3) ...
Konfigurowanie pakietu ssl-cert (1.1.2) ...
Konfigurowanie pakietu postfix (3.6.4-1ubuntu1.3) ...
Adding group 'postfix' (GID 121) ...
Done.
Adding system user 'postfix' (UID 115) ...
```

Kolejnym krokiem jest podanie ścieżki do repozytorium gitlab i jego instalacja

```
root@maciej:/home/maciej# curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.deb.sh | sudo bash
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 6865  100 6865    0     0  14292      0 --:--:-- --:--:-- --:--:-- 14302
Detected operating system as Ubuntu/jammy.
Checking for curl...
Detected curl...
Checking for gpg...
Detected gpg...
Running apt-get update... done.
Installing apt-transport-https... done.
Installing /etc/apt/sources.list.d/gitlab-ee.list...done.
Importing packagecloud gpg key... done.
Running apt-get update... done.

The repository is setup! You can now install packages.
root@maciej:/home/maciej#
```

Pobrane pakiety pozwolą na wprowadzenie komendy

```
root@maciej:/home/maciej# sudo apt-get install gitlab-ee
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
Zostaną zainstalowane następujące NOWE pakiety:
  gitlab-ee
0 aktualizowanych, 1 nowo instalowanych, 0 usuwanych i 4 nieaktualizowanych.
Konieczne pobranie 1 060 MB archiwów.
Po tej operacji zostanie dodatkowo użyte 3 359 MB miejsca na dysku.
Pobieranie:1 https://packages.gitlab.com/gitlab/gitlab-ee/ubuntu jammy/main amd64 gitlab-ee amd64 16.11.0-ee.0 [1 060 MB]
Pobrano 1 060 MB w 7min 45s (2 281 kB/s)
Wybieranie wcześniej niewybranego pakietu gitlab-ee.
(Odczytywanie bazy danych ... 75131 plików i katalogów obecnie zainstalowanych.)
Przgotowywanie do rozpakowania pakietu .../gitlab-ee_16.11.0-ee.0_amd64.deb ...
Rozpakowywanie pakietu gitlab-ee (16.11.0-ee.0) ...
Konfigurowanie pakietu gitlab-ee (16.11.0-ee.0) ...
It looks like GitLab has not been configured yet; skipping the upgrade script.
```

Konfigurujemy zaporę sieciową w celu zezwolenia na połączenia ssh wpisujemy komendy

Sudo ufw enable

Sudo ufw allow http

## Sudo ufw allow OpenSSH

## Następnie sprawdzamy porty

```
root@maciej:/home/maciej# ufw status
Status: active

To Action From
--
80/tcp ALLOW Anywhere
OpenSSH ALLOW Anywhere
80/tcp (v6) ALLOW Anywhere (v6)
OpenSSH (v6) ALLOW Anywhere (v6)

root@maciej:/home/maciej#
```

Udało się poprawnie skonfigurować zaporę teraz można w przeglądarce wpisać IP hosta poprzedzając go http://

Pierwsze logowanie następuje poprzez login root aby sprawdzić hasło należy wpisać następującą komendę `sudo cat /etc/gitlab/initial_root_password`

Następnie skonfigurujemy git lab na dockerze w tym celu trzeba utworzyć plik yml który zawiera podstawowy skrypt konfiguracyjny (porty, środowisko, adres serwera, usługi)

```
root@maciej:/etc/gitlab# more docker-compose.yml
version: '3.7'
services:
  web:
    image: 'gitlab/gitlab-ce:latest'
    restart: always
    hostname: 'localhost'
    container_name: gitlab-ce
    environment:
      GITLAB_OMNIBUS_CONFIG: |
        external_url 'http://localhost'
    ports:
      - '8080:80'
      - '8443:443'
    volumes:
      - '$GITLAB_HOME/config:/etc/gitlab'
      - '$GITLAB_HOME/logs:/var/log/gitlab'
      - '$GITLAB_HOME/data:/var/opt/gitlab'
    networks:
      - gitlab
  gitlab-runner:
    image: gitlab/gitlab-runner:alpine
    container_name: gitlab-runner
    restart: always
    depends_on:
      - web
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - '$GITLAB_HOME/gitlab-runner:/etc/gitlab-runner'
    networks:
      - gitlab
networks:
  gitlab:
    name: gitlab-network
```

Instalujemy docker compose jeżeli jeszcze nie jest on zainstalowany

```
root@maciej:/etc/gitlab# sudo apt-get install docker-compose
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
The following additional packages will be installed:
  python3-docker python3-dockerpty python3-dockerpty python3-dockerpty python3-dotenv python3-texttable python3-websocket
Zostaną zainstalowane następujące NOWE pakiety:
  docker-compose python3-docker python3-dockerpty python3-dockerpty python3-dockerpty python3-dotenv python3-texttable python3-websocket
0 aktualizowanych, 7 nowo instalowanych, 0 usuwanych i 4 nieaktualizowanych.
Konieczne pobranie 290 kB archiwów.
Po tej operacji zostanie dodatkowo użyte 1 545 kB miejsca na dysku.
Kontynuować? [T/n] T
Pobieranie:1 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-websocket all 1.2.3-1 [34,7 kB]
Pobieranie:2 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-docker all 5.0.3-1 [89,3 kB]
Pobieranie:3 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-dockerpty all 0.4.1-2 [11,1 kB]
Pobieranie:4 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-dockerpty all 0.6.2-4 [26,9 kB]
Pobieranie:5 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-dotenv all 0.19.2-1 [20,5 kB]
Pobieranie:6 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-texttable all 1.6.4-1 [11,4 kB]
Pobieranie:7 http://pl.archive.ubuntu.com/ubuntu jammy/universe amd64 docker-compose all 1.29.2-1 [95,8 kB]
Pobrano 290 kB w 1s (378 kB/s)
Wybieranie wcześniej niewybranego pakietu python3-websocket.
```

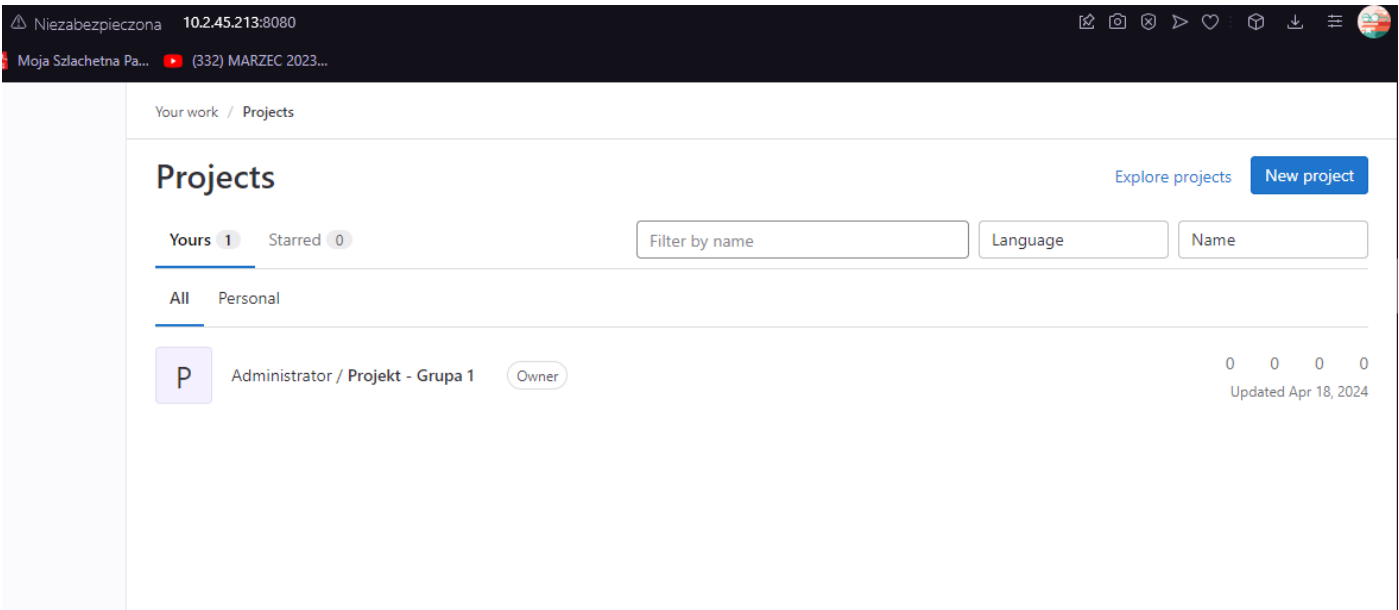
Uruchamiamy kontenery GitLaba poprzez docker-compose komendą

```
root@maciej:/etc/gitlab# sudo docker-compose up -d
WARNING: The GITLAB_HOME variable is not set. Defaulting to a blank string.
Creating network "gitlab-network" with the default driver
Pulling web (gitlab/gitlab-ce:latest)...
latest: Pulling from gitlab/gitlab-ce
7021d1b70935: Pull complete
84ce88366b95: Pull complete
e1c92fb77de1: Pull complete
719e91a15767: Pull complete
d680b9b016aa: Pull complete
f0027cb8bf12: Pull complete
8090ecb94c61: Pull complete
2d67bfd969b5: Pull complete
216d686bfe05: Extracting [=====] 1.205GB/1.205GB
```

Ważne jest odpowiednie ustawienie portów aby nie kolidowały ze sobą w przeciwnym razie system wykryje błąd należy ustawić w pliku yml porty, które nie są zajęte przez inne funkcje.

```
root@vdi desktop:/etc/gitlab# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
bfac6821cda    gitlab/gitlab-runner               "/usr/bin/dumb-init ..." 2 days ago    Up 38 hours   NAMES
1feef854abeb    gitlab/gitlab-runner:alpine        "/usr/bin/dumb-init ..." 2 days ago    Up 38 hours   new-gitlab-runner
756379bb98ca    gitlab/gitlab-ce:latest            "/assets/wrapper"         2 days ago    Up 38 hours   gitlab-runner
:::8080->80/tcp, 0.0.0.0:8443->443/tcp, :::8443->443/tcp  Up 38 hours   gitlab-ce
bd695683bbc2    gitlab/gitlab-ee:latest            "/assets/wrapper"         6 days ago    Up 38 hours   gitlab
tcp, 0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp  Up 38 hours   gitlab
```

Na koniec otwieramy gitlab poprzez wpisanie odpowiedniego adresu IP z portem





## KOLEJNY ETAP RUNNERY

```
root@vdi desktop:/etc/gitlab# docker run --rm -t -i gitlab/gitlab-runner -help
NAME:
  gitlab-runner - a GitLab Runner

USAGE:
  gitlab-runner [global options] command [command options] [arguments...]

VERSION:
  16.11.0 (91a27b2a)

AUTHOR:
  GitLab Inc. <support@gitlab.com>

COMMANDS:
  exec           execute a build locally
  list           List all configured runners
  run            run multi runner service
  register       register a new runner
  reset-token    reset a runner's token
  install        install service
  uninstall      uninstall service
  start          start service
  stop           stop service
  restart        restart service
  status         get status of a service
  run-single     start single runner
  unregister     unregister specific runner
  verify         verify all registered runners
  fleeting       manage fleeting plugins
  artifacts-downloader download and extract build artifacts (internal)
  artifacts-uploader   create and upload build artifacts (internal)
  cache-archiver       create and upload cache artifacts (internal)
  cache-extractor      download and extract cache artifacts (internal)
  cache-init           changed permissions for cache paths (internal)
  health-check         check health for a specific address
  read-logs            reads job logs from a file, used by kubernetes executor (internal)
  help, h             Shows a list of commands or help for one command
```

W naszym projekcie jeden z runnerów był domyślnie dołączony do kontenera powyższym poleceniem uruchamiamy kontener Docker wraz z obrazem gitlab-runner dzięki temu otrzymujemy pomoc dotyczącą tego programu.

Wcześniej polecenie zostało użyte i utworzony został kontener który od teraz pracuje w tle i w razie awarii lub problemów będzie resetowany.

```
root@vdi desktop:/etc/gitlab# docker run -d --name new-gitlab-runner --restart always gitlab/gitlab-runner
docker: Error response from daemon: Conflict. The container name "/new-gitlab-runner" is already in use by container "bfac6821cdaeb346c665fa926a23199b824d3e7fb3791d30ed75aafca98d62b". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
root@vdi desktop:/etc/gitlab#
```

Serwis git lab na platformie VDI działa bardzo powolnie i nie pozwala uruchomić poprawnie zakładki runners aby poznać token dzięki któremu rejestrujemy potrzebną instancję. Jednak nawet po wprowadzeniu odpowiedniego tokena po długim ładowaniu strony zostajemy przekierowani na stronę <https://docs.gitlab.com/runner/register/> okazuje się że autoryzacja tokenów gitlab w nowszych wersjach nie zakłada wprowadzania tokenów.

```
root@vdi desktop:/etc/gitlab# sudo gitlab-runner register
Runtime platform                                arch=amd64 os=linux pid=2279800 revision=81ab07f6 version=16.10.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
http://10.2.45.213
Enter the registration token:

```

Z powodu napotkanych problemów(powolne działanie serwisu gitlab na platformie VDI oraz zmieniony sposób autoryzacji tokenów zaprzestaliśmy dalszej konfiguracji runnerów.

```
root@vdi desktop:/etc/gitlab# docker run --rm -it -v /srv/gitlab-runner/config:/etc/gitlab-runner gitlab/gitlab-runner register
Runtime platform                                arch=amd64 os=linux pid=7 revision=91a27b2a version=16.11.0
Running in system-mode.

Created missing unique system ID                  system_id=r_96n77yRPBQKA
Enter the GitLab instance URL (for example, https://gitlab.com/):
```

## **Wnioski**

Gitlab to narzędzie dla programistów pozwalające pracować na projektami deweloperskimi. Umożliwia śledzenie zmian, zarządzanie repozytoriami oraz pozwala na automatyzację procesów przez CI/CD. Wszystkie wymienione funkcje pozwalają na optymalizację procesu związanego z tworzeniem nowych aplikacji oraz programów. Zespoły projektowe mogą w łatwy sposób dokonywać zmian i cały czas mieć dostęp do najnowszej wersji kodu źródłowego. Powyższe zadanie oprócz konfiguracji Gitlaba wymagało również poprawnej instalacji i konfiguracji platformy docker oraz narzędzia docker compose. Platforma Docker pozwala na tworzenie, zarządzanie oraz uruchamianie aplikacji w specjalnych kontenerach które można w prosty sposób przenosić na inne platformy. Docker compose pozwala zarządzać wieloma kontenerami jak i również sprawia że aplikacje można skonfigurować przez plik YAML. Podobnie jak docker usprawnia pracę nad kontenerami. Największą trudność na tym etapie sprawiło nam odpowiednie skonfigurowanie pliku YAML który pozwalał na połączenie się do gitlab przez odpowiedni port. Główną przeszkodą było znalezienie wolnego portu, który nie był aktualnie używany. Dodatkowo wiązało się to z konfiguracją zapory firewall oraz odpowiedniego włączenia portów http. Końcowo po zalogowaniu na Gitlab bardzo ważną kwestią jest zmiana hasła ponieważ jednorazowe hasło działa tylko przy pierwszym logowaniu i bez jego zmiany w późniejszych etapach mogą wystąpić problemy z zalogowaniem do projektu. Przechodząc do Runnerów służą one do różnych zadań które mogą być automatyczne. Umożliwiają budowanie, testowanie i wdrażanie oprogramowania. Runnery często wykorzystywane są w procesie CI/CD do wszelkich operacji automatyzacji. Realizacja tego etapu pozwoliła nam zrozumieć procesy instalacyjne wymienionych platform i dostarczyła wielu informacji dotyczących narzędzi programistycznych używanych do wdrażania aplikacji.