


SPRAWOZDANIE NR 3			
Nazwa ćwiczenia	ETAP 3 – KONFIGURACJA GITLABA		 <b>POLITECHNIKA BYDGOSKA</b> Wydział Telekomunikacji Informatyki i Elektrotechniki
Przedmiot	NARZĘDZIA PROGRAMISTYCZNE		
Student grupa	DORIAN ZASADA, JAKUB WEISS, MACIEJ KWIATKOWSKI, JULIAN FRISCHKE		
Data ćwiczeń	---	---	Data oddania sprawozdania

- 1. Celem tego etapu jest postanowienie i poprawne skonfigurowanie narzędzia gitlab na przygotowanym wcześniej systemie. Narzędzie to należy skonfigurować na Docker wykorzystując docker-compose.**
- 2. Przygotowanie**

Aby w ogóle ruszyć z konfigurowaniem gitlaba przy pomocy docker-compose, należy przygotować pod to odpowiednie miejsce. W naszym projekcie to miejsce znalazło się w /etc/gitlab. W folderze gitlab należy wtedy stworzyć plik docker-compose.yml. W pliku trzeba poprawnie skonfigurować każdy poszczególny element, gdyż to ten plik będzie brany pod uwagę przy budowie kontenera. Kontener budujemy za pomocą polecenia docker compose up -d. Natomiast uruchomione kontenery możemy zobaczyć po wpisaniu komendy docker ps. Pomimo wyświetlenia uruchomionych kontenerów to zostaniemy także poinformowani o „zdrowiu” naszego kontenera. Wiadomo, że najlepiej jakby kontener był „healthy”.

Problem był z hasłem. Przy niektórych konfiguracjach powstaje plik initial\_root\_password, które zawiera tymczasowe hasło. Niestety w naszym przypadku to się nie stało, musieliśmy to rozwiązać w inny dostępny sposób.

Należało na samym początku dostać się do basha.

```
docker exec -it gitlab-runner bash
```

później należało otworzyć konsolę

```
gitlab-rails console -e production
```

i po tym kroku nastąpiła zamiana hasła, dzięki komendom:

```
user = User.where(id: 1).first
```

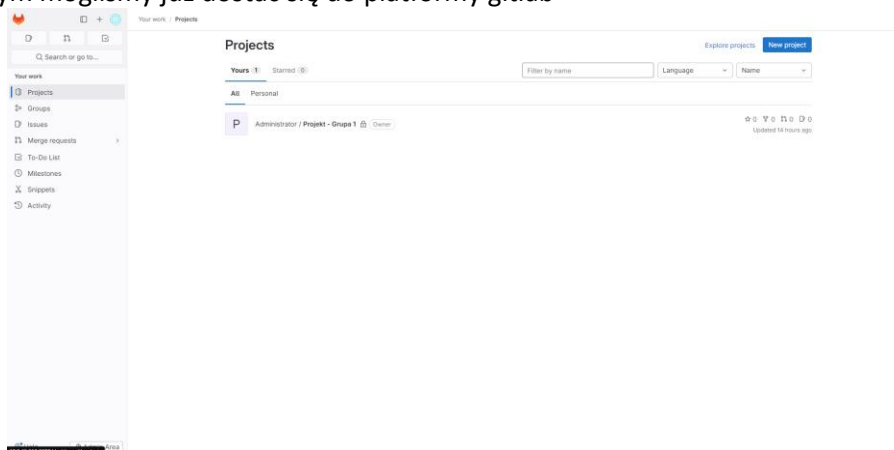
```
user.password = 'haslo'
```

```
user.password_confirmation = 'haslo'
```

```
user.save!
```

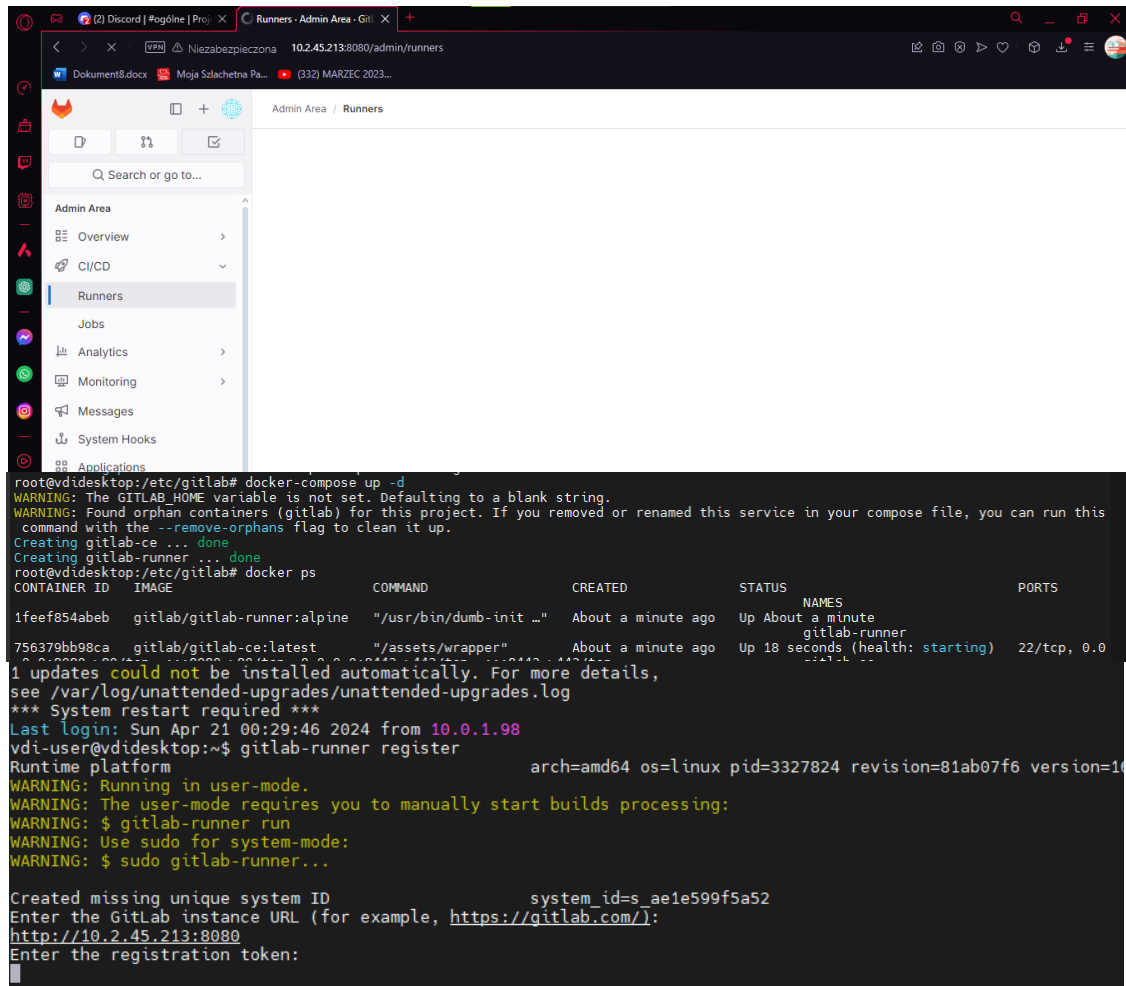
```
exit
```

Po tym mogliśmy już dostać się do platformy gitlab



I to tyle z tego etapu.

Na nasze nieszczęście nie udało nam się zautoryzować runnerów. Kłopot prawdopodobnie wynikał z problemów z maszyną VDI. Strona czasami się ładowała, czasami nie. Do zakładki z runnerami było ciężko się dostać. Okazało się, że GitLab nie wspiera już dłużej autoryzacji tokenami, więc trzeba było wymyślić inny sposób, który właśnie trzeba było uzyskać z pomocą strony.



The screenshot shows a web browser window with the GitLab Admin Area 'Runners' page. The browser's address bar shows the URL `10.2.45.213:8080/admin/runners`. The page has a sidebar with navigation links: Overview, CI/CD, Runners, Jobs, Analytics, Monitoring, Messages, and System Hooks. The main content area is currently blank.

Below the browser window, a terminal window displays the following commands and output:

```
root@vdidesktop:/etc/gitlab# docker-compose up -d
WARNING: The GITLAB_HOME variable is not set. Defaulting to a blank string.
WARNING: Found orphan containers (gitlab) for this project. If you removed or renamed this service in your compose file, you can run this
command with the --remove-orphans flag to clean it up.
Creating gitlab-ce ... done
Creating gitlab-runner ... done
root@vdidesktop:/etc/gitlab# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          NAMES
1feef854abeb   gitlab/gitlab-runner:alpine         "/usr/bin/dumb-init ..." About a minute ago Up About a minute gitlab-runner
756379bb98ca   gitlab/gitlab-ce:latest             "/assets/wrapper"        About a minute ago Up 18 seconds (health: starting) gitlab-ce
root@vdidesktop:/etc/gitlab# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          NAMES
1feef854abeb   gitlab/gitlab-runner:alpine         "/usr/bin/dumb-init ..." About a minute ago Up About a minute gitlab-runner
756379bb98ca   gitlab/gitlab-ce:latest             "/assets/wrapper"        About a minute ago Up 18 seconds (health: starting) gitlab-ce
root@vdidesktop:/etc/gitlab#
```

The terminal output continues with the following messages:

```
1 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
*** System restart required ***
Last login: Sun Apr 21 00:29:46 2024 from 10.0.1.98
vdi-user@vdidesktop:~$ gitlab-runner register
Runtime platform arch=amd64 os=linux pid=3327824 revision=81ab07f6 version=16.10.0
WARNING: Running in user-mode.
WARNING: The user-mode requires you to manually start builds processing:
WARNING: $ gitlab-runner run
WARNING: Use sudo for system-mode:
WARNING: $ sudo gitlab-runner...
```

The terminal output concludes with the following prompts:

```
Created missing unique system ID system_id=s_ae1e599f5a52
Enter the GitLab instance URL (for example, https://gitlab.com/):
http://10.2.45.213:8080
Enter the registration token:
```

### 3. Wnioski:

Podczas tego etapu nauczyliśmy się jak postawić nasz serwerowy gitlab. Czytać dokumentację oraz się do niej stosować. Praca z dockerem i gitlabem nie jest trudna, lecz wymaga dosyć dużo czasu to ogarnięcia co tak naprawdę się dzieje i co najważniejsze – po co.

Następnym etapem jest konfiguracja runnera, nie możemy się doczekać by zmierzyć się znowu z czymś nieznanym.