


SPRAWOZDANIE NR 2			
Nazwa ćwiczenia	Konfiguracja gitlab przez docker-compose		 POLITECHNIKA BYDGOSKA Wydział Telekomunikacji, Informatyki i Elektrotechniki
Przedmiot	Narzędzia Programistyczne		
Student grupa			
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

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

maciej@maciej-1-2:~$ sudo apt-get uptae
[sudo] hasło użytkownika maciej:
E: Nieprawidłowa operacja uptae
maciej@maciej-1-2:~$ sudo apt-get update
Stary:1 http://pl.archive.ubuntu.com/ubuntu mantic InRelease
Stary:2 http://security.ubuntu.com/ubuntu mantic-security InRelease
Stary:3 http://pl.archive.ubuntu.com/ubuntu mantic-updates InRelease
Stary:4 http://pl.archive.ubuntu.com/ubuntu mantic-backports InRelease
Pobieranie:5 http://pl.archive.ubuntu.com/ubuntu mantic/main Translation-pl [248
kB]
Pobieranie:6 http://pl.archive.ubuntu.com/ubuntu mantic/restricted Translation-p
l [596 B]
Pobieranie:7 http://pl.archive.ubuntu.com/ubuntu mantic/universe Translation-pl
[1 100 kB]
Pobieranie:8 http://pl.archive.ubuntu.com/ubuntu mantic/multiverse Translation-p
l [12,3 kB]
Pobrano 1 360 kB w 6s (225 kB/s)
Czytanie list pakietów... Gotowe
maciej@maciej-1-2:~$
```

```
6 kwi 23:38
Konta online
maciej@maciej-1-2: ~
Pobrano 1 360 kB w 6s (225 kB/s)
Czytanie list pakietów... Gotowe
maciej@maciej-1-2:~$ 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 git git-man liberror-perl pigz runc ubuntu-fan
Sugerowane pakiety:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
Zostaną zainstalowane następujące NOWE pakiety:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc
  ubuntu-fan
0 aktualizowanych, 9 nowo instalowanych, 0 usuwanych i 73 nieaktualizowanych.
Konieczne pobranie 74,7 MB archiwów.
Po tej operacji zostanie dodatkowo użyte 291 MB miejsca na dysku.
Kontynuować? [T/n] t
Pobieranie:1 http://pl.archive.ubuntu.com/ubuntu mantic/universe amd64 pigz amd6
4 2.6-1 [63,6 kB]
Pobieranie:2 http://pl.archive.ubuntu.com/ubuntu mantic/main amd64 bridge-utils
amd64 1.7.1-1ubuntu1 [34,9 kB]
Pobieranie:3 http://pl.archive.ubuntu.com/ubuntu mantic-updates/main amd64 runc
```

Następnie instalujemy połączenie SSH

```

gitlab_1 | [2024-04-06T23:46:39+00:00] INFO: rails_migration[gitlab-rails] sending run action to execute[clear the
gitlab-rails cache] (immediate)
gitlab_1 | Recipe: gitlab::gitlab-rails
maciej@maciej-1-2:/etc$ sudo nano docker-compose.yml
maciej@maciej-1-2:/etc$ sudo cat /etc/gitlab/initial_root_password
[sudo] hasło użytkownika maciej:
Niestety, proszę spróbować ponownie.
[sudo] hasło użytkownika maciej:
# WARNING: This value is valid only in the following conditions
# 1. If provided manually (either via 'GITLAB_ROOT_PASSWORD' environment variable or via 'gitlab_rails['initia
l_root_password']' setting in 'gitlab.rb', it was provided before database was seeded for the first time (usually, the
first reconfigure run).
# 2. Password hasn't been changed manually, either via UI or via command line.
#
# If the password shown here doesn't work, you must reset the admin password following https://docs.gitlab.com
/ee/security/reset_user_password.html#reset-your-root-password.

Password: KlgIR25yk29CAWRqvrWrtx37mKbzd2NxZWNh3d6G3c0=

# NOTE: This file will be automatically deleted in the first reconfigure run after 24 hours.
maciej@maciej-1-2:/etc$ hostname
maciej-1-2
maciej@maciej-1-2:/etc$ sudo apt-get install -y curl openssh-server ca-certificates tzdata perl
[sudo] hasło użytkownika maciej:
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
curl is already the newest version (8.2.1-1ubuntu3.3).
openssh-server is already the newest version (1:9.3p1-1ubuntu3.2).
ca-certificates is already the newest version (20230311ubuntu1).
tzdata is already the newest version (2024a-0ubuntu0.23.10).
perl is already the newest version (5.36.0-9ubuntu1.1).
0 aktualizowanych, 0 nowo instalowanych, 0 usuwanych i 0 nieaktualizowanych.
maciej@maciej-1-2:/etc$

```

Instalujemy pakiet pocztowy postfix

```
maciej@maciej-1-2:/etc$ sudo apt-get install -y postfix
```

Kolejnym krokiem jest podanie ścieżki do repozytorium gitlab i jego instalacja

```
maciej@maciej-1-2:/etc$ curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.deb.sh | sudo bas
h
```

Pobrane pakiety pozwolą na wprowadzenie komendy

```
maciej@maciej-1-2:/etc$ sudo apt-get install gitlab-ee
```

Finalnie po użyciu wymienionych komend narzędzie gitlab zostanie zainstalowane w systemie. Sprawdzamy to komendą `sudo gitlab-ctl status`

```

Warning: stdebqt: unable to open supervise/ok: access denied
maciej@maciej-1-2:/etc$ sudo gitlab-ctl status
run: alertmanager: (pid 31736) 27294s; run: log: (pid 31539) 27453s
run: gitaly: (pid 31723) 27297s; run: log: (pid 31008) 27854s
run: gitlab-exporter: (pid 31694) 27299s; run: log: (pid 31462) 27483s
run: gitlab-kas: (pid 31226) 27828s; run: log: (pid 31239) 27825s
run: gitlab-workhorse: (pid 31668) 27300s; run: log: (pid 31385) 27510s
run: logrotate: (pid 42205) 1345s; run: log: (pid 30936) 27866s
run: nginx: (pid 31678) 27300s; run: log: (pid 31407) 27503s
run: node-exporter: (pid 31687) 27300s; run: log: (pid 31446) 27494s
run: postgres-exporter: (pid 31745) 27293s; run: log: (pid 31569) 27444s
run: postgresql: (pid 31054) 27839s; run: log: (pid 31065) 27838s
run: prometheus: (pid 31708) 27298s; run: log: (pid 31511) 27464s
run: puma: (pid 31314) 27526s; run: log: (pid 31324) 27522s
run: redis: (pid 30963) 27861s; run: log: (pid 30972) 27860s
run: redis-exporter: (pid 31697) 27299s; run: log: (pid 31483) 27472s
run: sidekiq: (pid 31332) 27519s; run: log: (pid 31343) 27516s

```

Konfigurujemy zaporę sieciową w celu zezwolenia na połączenia ssh wpisujemy komendy

Sudo ufw enable

Sudo ufw allow http

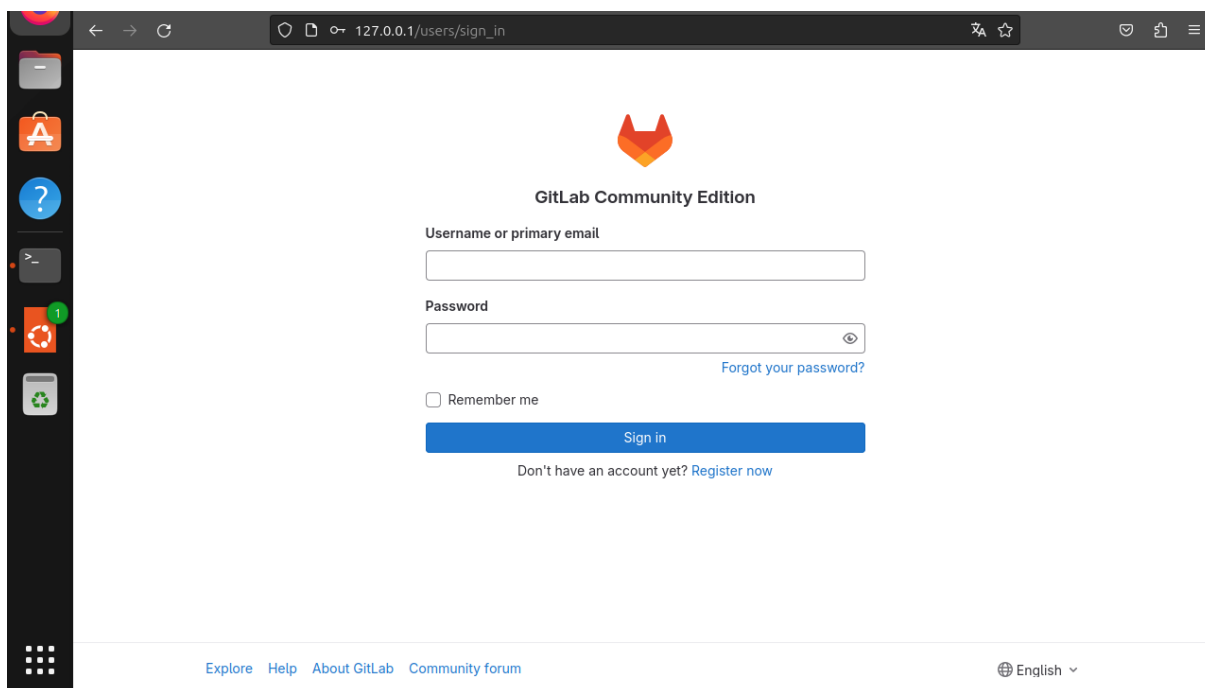
Sudo ufw allow OpenSSH

Następnie sprawdzamy porty

```
maciej@maciej-1-2:/etc$ sudo ufw status
Stan: aktywny

Do                Działanie  Z
--                -
80/tcp            ALLOW       Anywhere
443               ALLOW       Anywhere
OpenSSH           ALLOW       Anywhere
80/tcp (v6)       ALLOW       Anywhere (v6)
443 (v6)          ALLOW       Anywhere (v6)
OpenSSH (v6)      ALLOW       Anywhere (v6)
```

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`

```
maciej@maciej-1-2:/etc$ sudo cat /etc/gitlab/initial_root_password
[sudo] hasło użytkownika maciej:
# WARNING: This value is valid only in the following conditions
# 1. If provided manually (either via 'GITLAB_ROOT_PASSWORD' environment variable or via 'gitlab_rails['initial_root_password']' setting in 'gitlab.rb', it was provided before database was seeded for the first time (usually, the first reconfigure run).
# 2. Password hasn't been changed manually, either via UI or via command line.
#
# If the password shown here doesn't work, you must reset the admin password following https://docs.gitlab.com/ee/security/reset_user_password.html#reset-your-root-password.
Password: [REDACTED]
# NOTE: This file will be automatically deleted in the first reconfigure run after 24 hours.
maciej@maciej-1-2:/etc$
```

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

```
maciej@maciej-1-2: /etc$ cat docker-compose.yml
version: '3.8'

services:
  gitlab:
    image: 'gitlab/gitlab-ce:latest'
    restart: always
    hostname: 'gitlab.example.com'
    environment:
      GITLAB_OMNIBUS_CONFIG: |
        external_url 'http://gitlab.example.com'
    ports:
      - '9097:80'
      - '443:443'
      - '2222:22'
    volumes:
      - '/srv/gitlab/config:/etc/gitlab'
      - '/srv/gitlab/logs:/var/log/gitlab'
      - '/srv/gitlab/data:/var/opt/gitlab'

  postgresql:
    image: 'postgres:12'
    restart: always
    environment:
      POSTGRES_USER: 'gitlab'
      POSTGRES_PASSWORD: 'password'
      POSTGRES_DB: 'gitlabhq_production'
    volumes:
      - '/srv/gitlab/postgresql:/var/lib/postgresql'

  redis:
    image: 'redis:alpine'
    restart: always
```

Instalujemy docker compose jeżeli jeszcze nie jest on zainstalowany

```
maciej@maciej-1-2:/etc$ sudo apt install docker-compose
```

Uruchamiamy kontenery GitLaba poprzez docker-compose komendą

```
maciej@maciej-1-2:/etc$ sudo docker-compose up -d
```

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

Na koniec otwieramy gitlab poprzez wpisanie odpowiedniego adresu IP z portem