Instalacja Gitea w Dockerze

Typ: procedura **Dotyczy: Gitea 1.27.3, obraz docker.gitea.com/gitea:1.27.3**

Czas: ~30 min

Efekt: działający kontener Gitea z bazą PostgreSQL, dostępny po WWW na porcie 3000 i po SSH na porcie 2222, gotowy do pierwszej konfiguracji przez przeglądarkę.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS

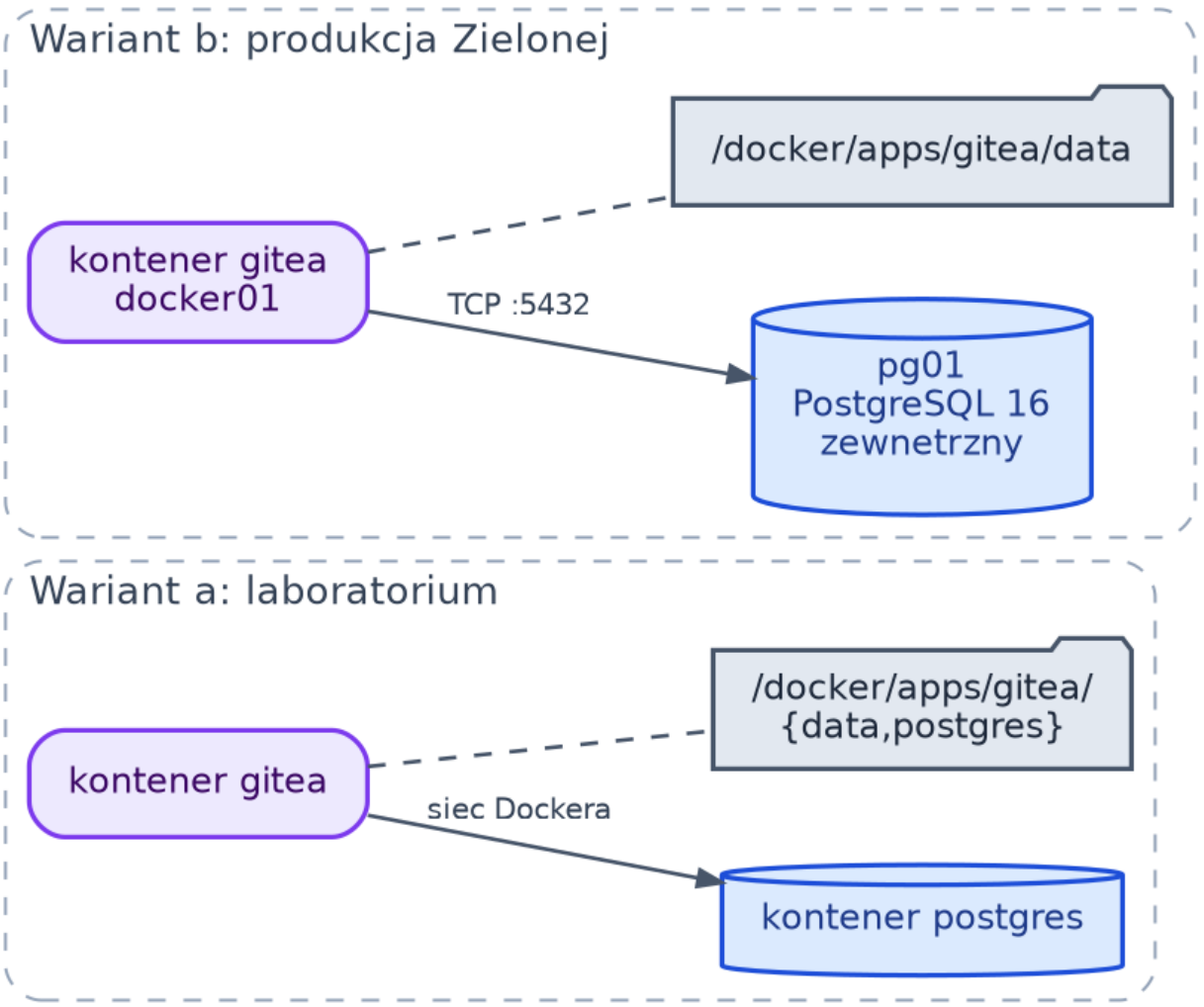
Zanim zaczniemy

Komendy wykonujemy z CLI bash jako user z uprawnieniami sudo albo root, na maszynie z zainstalowanym Dockerem i wtyczką Compose. Poniżej dwa warianty: laboratoryjny, z własnym kontenerem PostgreSQL obok Gitea, oraz produkcyjny Zielonej, z bazą na osobnym serwerze `pg01`. Wybieramy jeden, zależnie od tego, czy mamy już gdzieś działającą bazę PostgreSQL do współdzielenia.

> W naszym środowisku dane aplikacji trzymamy w `/docker/apps/<nazwa>`, a plik Compose w `/docker/compose/<nazwa>/compose.yaml`. To konwencja, nie wymóg Dockera: można ją zmienić, byle w całej firmie była jedna. Katalog `/docker` to osobny wolumen `vgDocker/lvDocker`, żeby pełny dysk aplikacji nie zatrzymał systemu.

Wymaganie	Uwaga
Docker Engine i wtyczka Compose	sprawdzamy `docker compose version`
Wolne porty `3000` i `2222` na hoście	port `2222`, nie `22`, patrz niżej
Wariant b: osiągalny `pg01` na porcie `5432`	baza tworzona ręcznie przed pierwszym startem
Wariant b: konto i baza założone w PostgreSQL na `pg01`	patrz krok 3 wariantu b

Uważaj: obraz zawsze przypinamy do konkretnej wersji (`1.27.3`), nigdy do `latest`. Aktualizacja to świadoma zmiana numeru w `compose.yaml`, nie przypadkowe pobranie nowszej wersji przy kolejnym `docker compose pull`.



Dwa warianty instalacji Gitea w Dockerze: laboratorium z własnym kontenerem PostgreSQL i produkcja Zielonej z bazą na pg01 (kliknięcie otwiera pełny rozmiar)

Przebieg

Wariant a: laboratorium, Gitea z własnym kontenerem PostgreSQL

Zakładamy katalogi na dane, zgodnie z konwencją ścieżek:

```
sudo mkdir -p /docker/apps/gitea/data /docker/apps/gitea/postgres
sudo mkdir -p /docker/compose/gitea
```

Plik `/docker/compose/gitea/.env` z hasłem do bazy, obok pliku `compose.yaml`, nie w nim samym:

```
GITEA_DB_PASSWORD=HasloDoBazy
```

Plik `/docker/compose/gitea/compose.yaml`:

```
services:
  server:
    image: docker.gitea.com/gitea:1.27.3
    container_name: gitea
    restart: always
    environment:
      - USER_UID=1000
      - USER_GID=1000
      - GITEA__database__DB_TYPE=postgres
      - GITEA__database__HOST=db:5432
      - GITEA__database__NAME=gitea
      - GITEA__database__USER=gitea
      - GITEA__database__PASSWD=${GITEA_DB_PASSWORD}
    volumes:
      - /docker/apps/gitea/data:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "3000:3000"
      - "2222:22"
    depends_on:
      - db

  db:
    image: postgres:16
```

```
    container_name: gitea-postgres
    restart: always
    environment:
      - POSTGRES_USER=gitea
      - POSTGRES_PASSWORD=${GITEA_DB_PASSWORD}
      - POSTGRES_DB=gitea
    volumes:
      - /docker/apps/gitea/postgres:/var/lib/postgresql/data
```

Pierwsze uruchomienie:

```
cd /docker/compose/gitea
docker compose up -d
```

Wariant b: produkcja Zielonej, baza na zewnętrznym pg01

Katalog na dane potrzebny jest tylko dla samej Gitea, bez podkatalogu na bazę:

```
sudo mkdir -p /docker/apps/gitea/data
sudo mkdir -p /docker/compose/gitea
```

Zanim uruchomimy kontener, zakładamy bazę i użytkownika na `pg01`. Łączymy się tam jako administrator PostgreSQL:

```
psql -h pg01 -U postgres
```

W sesji `psql` tworzymy bazę i konto aplikacji:

```
CREATE DATABASE gitea;
CREATE USER gitea WITH ENCRYPTED PASSWORD 'HasloDoBazy';
GRANT ALL PRIVILEGES ON DATABASE gitea TO gitea;
ALTER DATABASE gitea OWNER TO gitea;
```

Plik `/docker/compose/gitea/.env`:

```
GITEA_DB_PASSWORD=HasloDoBazy
```

Plik `/docker/compose/gitea/compose.yaml`:

```
services:
  server:
    image: docker.gitea.com/gitea:1.27.3
    container_name: gitea
    restart: always
    environment:
      - USER_UID=1000
      - USER_GID=1000
      - GITEA__database__DB_TYPE=postgres
      - GITEA__database__HOST=pg01:5432
      - GITEA__database__NAME=gitea
      - GITEA__database__USER=gitea
      - GITEA__database__PASSWD=${GITEA_DB_PASSWORD}
    volumes:
      - /docker/apps/gitea/data:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "3000:3000"
      - "2222:22"
```

Wartości zmiennych środowiskowych w postaci `GITEA__sekcja__KLUCZ` odpowiadają dokładnie sekcjom i kluczom pliku `app.ini`, opisanego na stronie Pierwsze uruchomienie: kreator, administrator, reverse proxy: `GITEA__database__HOST` nadpisuje `HOST` w sekcji `[database]`, i tak dalej. Pierwsze uruchomienie:

```
cd /docker/compose/gitea
docker compose up -d
```

Dlaczego port 2222, nie 22

Kontener nasłuchuje wewnątrz na standardowym porcie SSH `22`, ale na hoście mapujemy go na `2222`. Port `22` hosta jest zwykle już zajęty przez systemowy `sshd`, którym administrujemy samą maszyną `docker01`, niezależnie od Gitea. Rozdzielenie portów pozwala też ograniczyć na zaporze dostęp do administracyjnego SSH hosta osobno od SSH używanego przez programistów do operacji `git clone`/`push`/`pull`.

Logi pierwszego startu

Sprawdzamy, czy kontener wstał poprawnie i połączył się z bazą:

```
docker compose logs -f server
```

Przykład:

```
gitea  | 2026/09/27 10:02:14 ...s/setting/setting.go:320:NewContext() [I] Custom config:
/data/gitea/conf/app.ini
gitea  | 2026/09/27 10:02:15 ...ute/routes/common/db.go:32 [I] ORM engine initialization
successful!
gitea  | 2026/09/27 10:02:15 ...ute/setting/git.go:213:newGit() [I] Git Version: 2.45.0
gitea  | 2026/09/27 10:02:16 ...able/server/main.go:120 [I] Listen: http://0.0.0.0:3000
gitea  | 2026/09/27 10:02:16 ...ute/graceful/server.go:75:NewServer() [I] Starting new web
server: tcp:0.0.0.0:3000 on PID: 1
```

Weryfikacja

Sprawdzamy, że kontenery działają:

```
docker compose ps
```

Przykład:

```
NAME              IMAGE                              STATUS
gitea             docker.gitea.com/gitea:1.27.3      Up 2 minutes
gitea-postgres    postgres:16                        Up 2 minutes
```

Sprawdzamy, że interfejs WWW odpowiada, zanim otworzymy go w przeglądarce:

```
curl -I http://localhost:3000
```

Przykład:

```
HTTP/1.1 200 OK
Set-Cookie: i_like_gitea=...
Set-Cookie: _csrf=...
```

🖼 ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: pierwszy ekran otwarty pod `http://docker01:3000`, czyli kreator instalacji Gitea („Install Gitea") z widoczną sekcją „Database Settings" i typem bazy „PostgreSQL".

Wycofanie zmian

Zatrzymanie i usunięcie kontenerów, bez kasowania danych z wolumenów:

```
docker compose down
```

Pełne usunięcie razem z danymi, tylko gdy naprawdę chcemy zacząć od zera:

```
docker compose down -v
sudo rm -rf /docker/apps/gitea
```

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
Kontener `server` restartuje się w pętli	baza jeszcze nie wystartowała (wariant a) albo `pg01` nieosiągalny (wariant b)	`docker compose logs db` albo sprawdzamy `telnet pg01 5432` z hosta `docker01`
`curl -I http://localhost:3000` zwraca `Connection refused`	kontener jeszcze się uruchamia albo port zajęty przez inną usługę	`docker compose logs server`, `ss -tlnp`
Błąd `password authentication failed for user gitea`	hasło w `.env` inne niż w bazie na `pg01`	poprawiamy hasło w `.env` albo w bazie przez `ALTER USER gitea WITH PASSWORD`
Push po SSH nie działa, mimo że WWW działa	firewall blokuje port `2222` albo klient łączy się na `22`	sprawdzamy `firewall-cmd --list-ports` i adres w konfiguracji klienta Git

Powiązane: Instalacja Gitea bez Dockera · Pierwsze uruchomienie: kreator, administrator, reverse proxy · Konwencja Dockera w Zielonej

Źródła: docs.gitea.com, rozdział „Installation with Docker" · docs.gitea.com, rozdział „Config Cheat Sheet" (zmienne `GITEA__`)

Revision #33
Created 2026-09-27 10:32:03 UTC by Tomasz Konieczny
Updated 2026-10-26 14:17:27 UTC by Tomasz Konieczny