Instalacja serwera Zabbix w Dockerze

Typ: procedura

Dotyczy: Zabbix 7.0.31 LTS, obrazy zabbix/zabbix-server-pgsql, zabbix/zabbix-web-nginx-pgsql, zabbix/zabbix-agent2 (sprawdzone na Docker Hub 28.09.2026)

Czas: ~35 min

Efekt: działający serwer, frontend i agent Zabbixa w kontenerach, z bazą PostgreSQL w wariancie laboratoryjnym albo na `pg01` w wariancie produkcyjnym, gotowe do pierwszego logowania.

Skróty użyte na tej stronie (2)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
LTS	Long Term Support: wersja z długim wsparciem poprawkami

Zanim zaczniemy

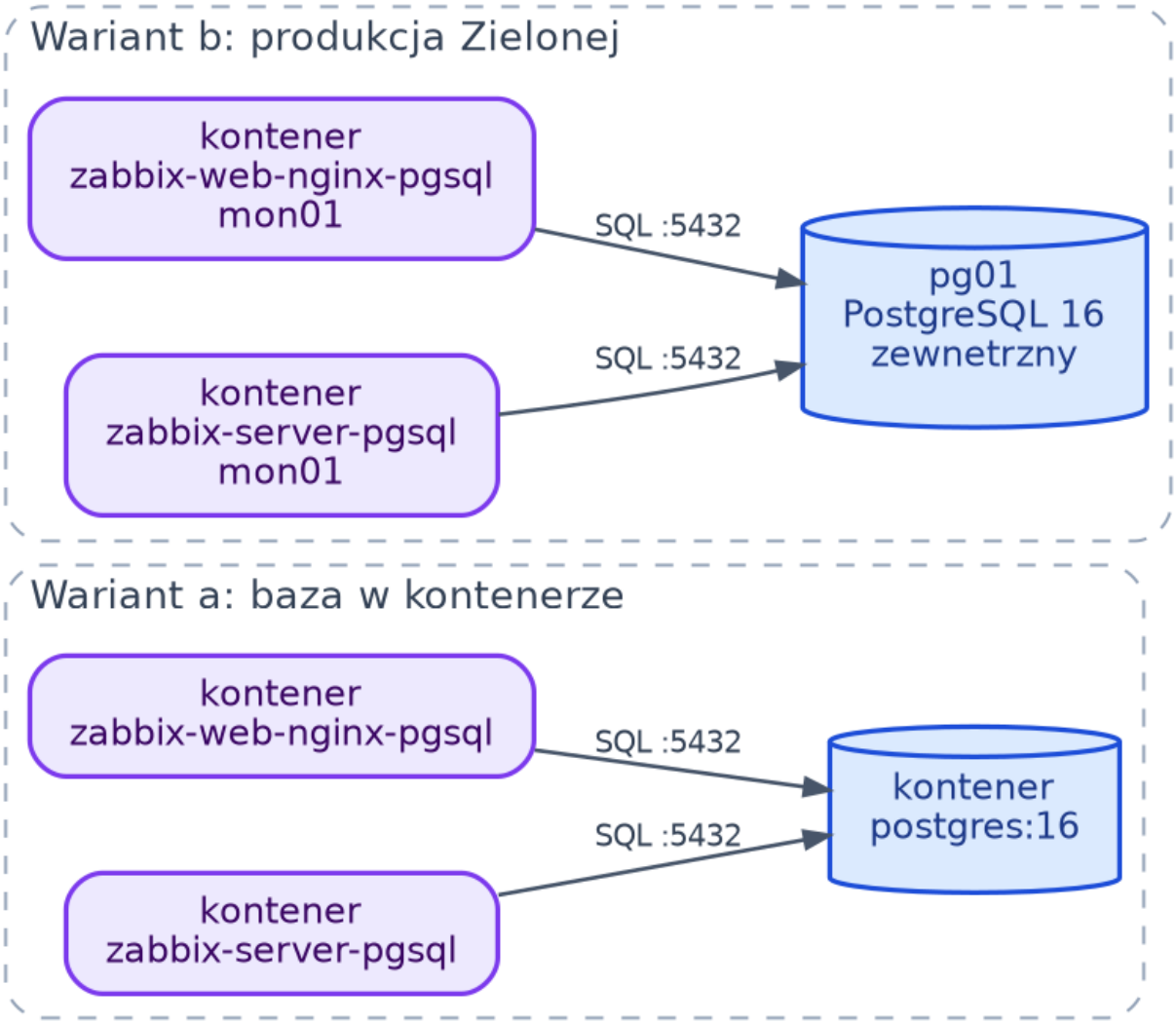
Komendy wykonujemy z CLI bash jako user z uprawnieniami sudo albo root, na maszynie z zainstalowanym Dockerem i wtyczką Compose. Poniżej dwa warianty: laboratoryjny, z własnym kontenerem PostgreSQL obok Zabbixa, oraz produkcyjny Zielonej, z bazą na osobnym serwerze `pg01`. Wybieramy jeden, zależnie od 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`
Wolny port `80` na hoście dla frontendu	kontener nasłuchuje wewnątrz na `8080`, mapujemy na `80`

Wymaganie	Uwaga
Wolny port `10051` na hoście	port serwera, potrzebny do sprawdzeń aktywnych z innych hostów
Wariant b: osiągalny `pg01` na porcie `5432`	baza tworzona ręcznie przed pierwszym startem

Uważaj: obraz zawsze przypinamy do konkretnej wersji (`7.0.31-ubuntu`), nigdy do `latest`. Aktualizacja to świadoma zmiana numeru w `compose.yaml`, nie przypadkowe pobranie nowszej wersji przy kolejnym `docker compose pull`. Wersja `7.0` to aktualna gałąź LTS Zabbixa; kolejna LTS, `8.0`, w chwili sprawdzania nie miała jeszcze wydania stabilnego, więc trzymamy się `7.0`.



Wariant a trzyma bazę we własnym kontenerze, wariant b łączy się do PostgreSQL na pg01 (kliknięcie otwiera pełny rozmiar)

Przebieg

Wariant a: laboratorium, Zabbix z własnym kontenerem PostgreSQL

Zakładamy katalogi na dane, zgodnie z konwencją ścieżek:

```
sudo mkdir -p /docker/apps/zabbix/server /docker/apps/zabbix/postgres
sudo mkdir -p /docker/compose/zabbix
```

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

```
ZABBIX_DB_PASSWORD=HasloDoBazy
```

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

```
services:
  server:
    image: zabbix/zabbix-server-pgsql:7.0.31-ubuntu
    container_name: zabbix-server
    restart: always
    environment:
      - DB_SERVER_HOST=db
      - POSTGRES_DB=zabbix
      - POSTGRES_USER=zabbix
      - POSTGRES_PASSWORD=${ZABBIX_DB_PASSWORD}
    volumes:
      - /docker/apps/zabbix/server:/var/lib/zabbix/export
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "10051:10051"
    depends_on:
      - db

  web:
    image: zabbix/zabbix-web-nginx-pgsql:7.0.31-ubuntu
```

```
    container_name: zabbix-web
    restart: always
    environment:
      - DB_SERVER_HOST=db
      - POSTGRES_DB=zabbix
      - POSTGRES_USER=zabbix
      - POSTGRES_PASSWORD=${ZABBIX_DB_PASSWORD}
      - ZBX_SERVER_HOST=server
      - PHP_TZ=Europe/Warsaw
    ports:
      - "80:8080"
    depends_on:
      - server
      - db

  agent:
    image: zabbix/zabbix-agent2:7.0.31-ubuntu
    container_name: zabbix-agent2
    restart: always
    environment:
      - ZBX_HOSTNAME=mon01
      - ZBX_SERVER_HOST=server
    volumes:
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    depends_on:
      - server

  db:
    image: postgres:16
    container_name: zabbix-postgres
    restart: always
    environment:
      - POSTGRES_USER=zabbix
      - POSTGRES_PASSWORD=${ZABBIX_DB_PASSWORD}
      - POSTGRES_DB=zabbix
    volumes:
      - /docker/apps/zabbix/postgres:/var/lib/postgresql/data
```

Pierwsze uruchomienie:

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

Wariant b: produkcja Zielonej, baza na zewnętrznym pg01

Katalog na dane potrzebny jest tylko dla samego Zabbixa, bez podkatalogu na bazę:

```
sudo mkdir -p /docker/apps/zabbix/server
sudo mkdir -p /docker/compose/zabbix
```

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

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

```
ZABBIX_DB_PASSWORD=HasloDoBazy
```

Plik `/docker/compose/zabbix/compose.yaml`: identyczny jak w wariancie a, tylko bez usługi `db` i z `DB_SERVER_HOST=pg01` w usługach `server` i `web` zamiast `DB_SERVER_HOST=db`:

```
services:
  server:
    image: zabbix/zabbix-server-pgsql:7.0.31-ubuntu
    container_name: zabbix-server
    restart: always
    environment:
      - DB_SERVER_HOST=pg01
      - POSTGRES_DB=zabbix
      - POSTGRES_USER=zabbix
```

```
      - POSTGRES_PASSWORD=${ZABBIX_DB_PASSWORD}
    volumes:
      - /docker/apps/zabbix/server:/var/lib/zabbix/export
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "10051:10051"

  web:
    image: zabbix/zabbix-web-nginx-pgsql:7.0.31-ubuntu
    container_name: zabbix-web
    restart: always
    environment:
      - DB_SERVER_HOST=pg01
      - POSTGRES_DB=zabbix
      - POSTGRES_USER=zabbix
      - POSTGRES_PASSWORD=${ZABBIX_DB_PASSWORD}
      - ZBX_SERVER_HOST=server
      - PHP_TZ=Europe/Warsaw
    ports:
      - "80:8080"
    depends_on:
      - server

  agent:
    image: zabbix/zabbix-agent2:7.0.31-ubuntu
    container_name: zabbix-agent2
    restart: always
    environment:
      - ZBX_HOSTNAME=mon01
      - ZBX_SERVER_HOST=server
    volumes:
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    depends_on:
      - server
```

`DB_SERVER_HOST` to jedyna zmienna, która różni się między wariantami: w wariancie a wskazuje na nazwę usługi `db` w tej samej sieci Dockera, w wariancie b na nazwę serwera `pg01` osiągalnego w sieci Zielonej. Pierwsze uruchomienie:

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

Logi pierwszego startu

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

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

Przykład:

```
zabbix-server  | 2026/09/28 08:11:04 ...include/zbxdbschema.c:... database version 07000000
zabbix-server  | 2026/09/28 08:11:04 ...server.c:... server #0 started [main process]
zabbix-server  | 2026/09/28 08:11:04 ...dbsync.c:... syncing configuration cache
```

Weryfikacja

Sprawdzamy, że kontenery działają:

```
docker compose ps
```

Przykład:

```
NAME             IMAGE                                          STATUS
zabbix-server    zabbix/zabbix-server-pgsql:7.0.31-ubuntu       Up 2 minutes
zabbix-web       zabbix/zabbix-web-nginx-pgsql:7.0.31-ubuntu    Up 2 minutes
zabbix-agent2    zabbix/zabbix-agent2:7.0.31-ubuntu             Up 2 minutes
```

Sprawdzamy, że frontend odpowiada, zanim otworzymy go w przeglądarce:

```
curl -I http://localhost:80
```

Przykład:

```
HTTP/1.1 302 Found
Location: index.php
```

Otwieramy `http://mon01` w przeglądarce i logujemy się domyślnym kontem administratora:

```
Login: Admin
Password: zabbix
```

> Zaraz po pierwszym zalogowaniu zmieniamy hasło konta `Admin`, zanim zrobimy cokolwiek innego: `Users → Users → Admin → Change password`. Domyślne hasło `zabbix` jest publicznie znane z dokumentacji, więc zostawienie go choćby na chwilę to otwarta furtka do pełnej administracji Zabbiksem.

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran logowania Zabbixa pod `http://mon01`, a zaraz po nim ekran `Users → Users → Admin` z otwartym formularzem zmiany hasła.

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/zabbix
```

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 `mon01`
`curl -I http://localhost:80` zwraca `Connection refused`	kontener `web` jeszcze się uruchamia albo port zajęty przez inną usługę	`docker compose logs web`, `ss -tlnp \

Objaw	Przyczyna	Naprawa
Frontend wyświetla błąd połączenia z bazą	inne hasło w `.env` niż w bazie na `pg01`	poprawiamy hasło w `.env` albo w bazie przez `ALTER USER zabbix WITH PASSWORD`
Nowe hosty pokazują status niedostępny mimo działającego serwera	port `10051` niedostępny z zewnątrz kontenera	sprawdzamy mapowanie portu w `compose.yaml` i firewalld na hoście `mon01`

Powiązane: Instalacja serwera Zabbix z pakietów na Rocky Linux 9 · Architektura Zabbixa · Konwencja Dockera w Zielonej

Źródła: Zabbix Docker images · zabbix/zabbix-server-pgsql, Docker Hub · zabbix/zabbix-web-nginx-pgsql, Docker Hub · Zabbix Life Cycle and Release Policy

---

Revision #23
Created 2026-09-28 16:57:49 UTC by Tomasz Konieczny
Updated 2026-10-26 14:09:40 UTC by Tomasz Konieczny