GLPI w Zielonej: zgłoszenia, SLA i katalog usług na docker01

Poziom: średnio

Dotyczy: GLPI 12.0.0 (oficjalny obraz glpi/glpi:12.0.0), MariaDB 11.4.13, docker01, Traefik, MailCow

Czas: ~60 min

GLPI to darmowe narzędzie do zgłoszeń, w którym zapisujemy wszystko, o czym mówi ta książka: incydenty, wnioski, problemy, zmiany, katalog usług i umowy SLA. Strona stawia GLPI w kontenerze na `docker01` według konwencji Zielonej, konfiguruje go pod nasze standardy, podłącza skrzynkę `it@zielona.example` z MailCow i opisuje kopię i aktualizację. Na końcu jest krótkie porównanie z Jira Service Management, Zammadem i iTopem.

Skróty użyte na tej stronie (16)

Skrót	Co znaczy
AGPL	Affero General Public License: licencja wolnego oprogramowania, na której MongoDB było wydawane przed przejściem na SSPL w 2018 roku
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
GLPI	Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01
GPL	General Public License: otwarta licencja oprogramowania, na której opierają się MySQL Community i MariaDB
IMAP	Internet Message Access Protocol: protokół, którym klient pocztowy pobiera wiadomości z serwera, zostawiając je tam zamiast kasować
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
OLA	Operational Level Agreement: umowa operacyjna wewnątrz dostawcy usługi, np. między zespołem IT a zespołem programistów, na której stoi SLA

Skrót	Co znaczy
PHP	PHP: Hypertext Preprocessor, język programowania, w którym napisany jest frontend Zabbixa i wiele innych aplikacji webowych
README	Read Me: plik tekstowy z opisem projektu, wykresu lub szablonu, czytany jako pierwszy
SLA	Service Level Agreement: umowa o poziomie usług, np. 99,9% dostępności
SMTP	Simple Mail Transfer Protocol: protokół wysyłania poczty
TLS	Transport Layer Security: szyfrowanie połączeń, np. w HTTPS
UID	User ID: numer użytkownika w Linuksie
VPN	Virtual Private Network: szyfrowany tunel do sieci firmowej
WMS	Warehouse Management System: system zarządzania magazynem

Fiszka

Komendy wykonujemy na `docker01` w CLI bash jako użytkownik z uprawnieniami sudo, w katalogu `/docker/compose/glpi`, chyba że karta mówi inaczej. Ścieżki w interfejsie GLPI podajemy po angielsku, tak jak w dokumentacji; dokumentacja raz nazywa główne menu Setup, a raz Configuration, więc w razie rozbieżności szukamy po nazwie pozycji.

> W naszym środowisku dane aplikacji trzymamy w `/docker/apps/glpi`, a plik Compose w `/docker/compose/glpi/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.

◫◫ ZMIENIA SYSTEM

Założenie katalogów: mkdir -p

```
sudo mkdir -p /docker/apps/glpi/glpi /docker/apps/glpi/plugins /docker/apps/glpi/mariadb
/docker/compose/glpi
```

Katalog `glpi` trzyma konfigurację, załączniki, logi i wtyczki z Marketplace (w kontenerze `/var/glpi`, wtyczki w `/var/glpi/marketplace`), `plugins` wtyczki wgrane ręcznie (w kontenerze `/var/www/glpi/plugins`), a `mariadb` pliki bazy.

◫◫ **mkdir** = make directory: utwórz katalog

⚠ OSTROŻNIE · NIEODWRACALNE

Sprawdzenie użytkownika w obrazie: docker run --entrypoint id

```
docker run --rm --entrypoint id glpi/glpi:12.0.0
```

Obraz GLPI działa jako użytkownik `www-data`, a skrypt startowy przerywa pracę, jeśli nie może pisać do `/var/glpi`. Komenda pokazuje UID tego użytkownika, który potrzebny jest w następnej karcie; w obrazach opartych na Debianie (a ten bazuje na oficjalnym obrazie PHP z Apache) jest to zwykle 33.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **UID** = User ID: numer użytkownika w Linuksie · **PHP** = PHP: Hypertext Preprocessor, język programowania, w którym napisany jest frontend Zabbixa i wiele innych aplikacji webowych · **docker** = narzędzie do kontenerów

⚠ ZMIENIA SYSTEM

Nadanie właściciela katalogowi danych: chown

```
sudo chown -R UID:GID /docker/apps/glpi/glpi /docker/apps/glpi/plugins
```

> (gdzie `UID:GID` to wartości z poprzedniej karty, np. `33:33`)

Bez tego kontener zatrzyma się z komunikatem, że katalog nie jest zapisywalny. Katalogu `mariadb` nie ruszamy: obraz MariaDB sam ustawia właściciela przy pierwszym starcie.

UID = User ID: numer użytkownika w Linuksie · **GID** = Group Identifier: numer grupy systemowej, drugi obok UID element decydujący o prawach dostępu do pliku · **chown** = change owner: zmień właściciela

PLIK KONFIGURACYJNY

Plik z hasłami: .env

```
GLPI_WERSJA=12.0.0
MARIADB_WERSJA=11.4.13
MARIADB_ROOT_PASSWORD=HasloRootaBazy
MARIADB_DATABASE=glpi
MARIADB_USER=glpi
```

```
MARIADB_PASSWORD=HasloUzytkownikaGlpi
```

Wersje obrazów trzymamy w `.env`, żeby aktualizacja była zmianą jednej linii. Plik ma prawa `600` (`sudo chmod 600 /docker/compose/glpi/.env`).

PLIK KONFIGURACYJNY

Plik Compose: compose.yaml

```
services:
  glpi:
    image: glpi/glpi:${GLPI_WERSJA}
    container_name: glpi
    restart: unless-stopped
    depends_on:
      db:
        condition: service_healthy
    environment:
      GLPI_DB_HOST: db
      GLPI_DB_PORT: "3306"
      GLPI_DB_NAME: ${MARIADB_DATABASE}
      GLPI_DB_USER: ${MARIADB_USER}
      GLPI_DB_PASSWORD: ${MARIADB_PASSWORD}
      TZ: Europe/Warsaw
    volumes:
      - /docker/apps/glpi/glpi:/var/glpi
      - /docker/apps/glpi/plugins:/var/www/glpi/plugins
      - /docker/compose/glpi/strefa-czasowa.ini:/usr/local/etc/php/conf.d/strefa-
czasowa.ini:ro
    networks:
      - default
      - proxy
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.glpi.rule=Host(`pomoc.zielona.example`)"
      - "traefik.http.routers.glpi.entrypoints=websecure"
      - "traefik.http.routers.glpi.tls=true"
      - "traefik.http.services.glpi.loadbalancer.server.port=80"
```

```
  db:
    image: mariadb:${MARIADB_WERSJA}
    container_name: glpi-mariadb
    restart: unless-stopped
    environment:
      MARIADB_ROOT_PASSWORD: ${MARIADB_ROOT_PASSWORD}
      MARIADB_DATABASE: ${MARIADB_DATABASE}
      MARIADB_USER: ${MARIADB_USER}
      MARIADB_PASSWORD: ${MARIADB_PASSWORD}
      TZ: Europe/Warsaw
    volumes:
      - /docker/apps/glpi/mariadb:/var/lib/mysql
    healthcheck:
      test: ["CMD", "healthcheck.sh", "--connect", "--innodb_initialized"]
      interval: 10s
      timeout: 5s
      retries: 10
    networks:
      - default

networks:
  default:
  proxy:
    external: true
```

Obraz `glpi/glpi` deklaruje jeden wolumen, `/var/glpi`, z katalogami `config`, `files`, `logs` i `marketplace`, więc wtyczki instalowane z Marketplace są w nim od GLPI 11 (w GLPI 10 Marketplace leżał w `/var/www/glpi/marketplace` i potrzebował osobnego montowania). Katalog `/var/www/glpi/plugins` leży poza wolumenem: wtyczka wgrana tam ręcznie bez montowania znika przy każdym odtworzeniu kontenera, na przykład przy aktualizacji obrazu. Dlatego montujemy go z `/docker/apps/glpi/plugins`, nawet jeśli dziś jest pusty.

GLPI nie publikuje żadnego portu na hoście: ruch przychodzi wyłącznie przez Traefika w sieci `proxy`, tak jak w każdej aplikacji na `docker01`. Baza jest tylko w sieci domyślnej projektu, niewidoczna dla innych kontenerów.

□□ **GLPI** = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **CMD** = skrót od polecenia command, dawna nazwa wiersza poleceń Windows sprzed PowerShella

□□ PLIK KONFIGURACYJNY

Strefa czasowa PHP: strefa-czasowa.ini

```
date.timezone = Europe/Warsaw
```

Plik leży w `/docker/compose/glpi/strefa-czasowa.ini` i jest montowany do katalogu konfiguracji PHP w obrazie. Bez niego godziny w zgłoszeniach i liczenie SLA mogą się rozjechać z kalendarzem pracy.

□□ **PHP** = PHP: Hypertext Preprocessor, język programowania, w którym napisany jest frontend Zabbixa i wiele innych aplikacji webowych · **SLA** = Service Level Agreement: umowa o poziomie usług, np. 99,9% dostępności

□□ ZMIENIA SYSTEM

Wpis DNS dla adresu GLPI: Add-DnsServerResourceRecordCName

```
Add-DnsServerResourceRecordCName -ZoneName "zielona.example" -Name "pomoc" -
HostNameAlias "docker01.ad.zielona.example" -ComputerName "dc01"
```

Adres `pomoc.zielona.example` wskazuje na `docker01`, gdzie Traefik kieruje ruch do kontenera po nazwie hosta. Certyfikat dla tej nazwy wystawiamy tak jak dla innych aplikacji za Traefikiem.

□□ **DNS** = Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP · **GLPI** = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01

□□ ZMIENIA SYSTEM

Uruchomienie kontenerów: docker compose up -d

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

Przy pierwszym starcie obraz sam instaluje bazę GLPI, bo w zmiennych są wszystkie dane połączenia. Przy kolejnych startach sprawdza, czy baza wymaga aktualizacji, i wykonuje ją automatycznie.

□□ **GLPI** = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **cd** = change directory: przejdź do katalogu · **docker** = narzędzie do kontenerów

TYLKO ODCZYT

Podgląd instalacji: docker compose logs

```
docker compose logs -f glpi
```

Czekamy na powitanie z informacją, że instalacja się zakończyła. Do tego czasu strona w przeglądarce może zwracać błąd.

Przykład końca dziennika po pierwszym starcie:

```
glpi  | GLPI is not installed. but auto-install is enabled. Starting installation.
glpi  | Please wait until you see the greeting, this may take a minute...
glpi  | GLPI installation completed successfully!
```

docker = narzędzie do kontenerów

TYLKO ODCZYT

Sprawdzenie wymagań: bin/console system:check_requirements

```
docker compose exec glpi bin/console system:check_requirements
```

Lista wymagań GLPI: wersja PHP, rozszerzenia, wersja bazy (GLPI 12 wymaga co najmniej MariaDB 10.11), prawa do katalogów, strefy czasowe. Każdy wiersz z ostrzeżeniem poprawiamy przed oddaniem GLPI użytkownikom.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **PHP** = PHP: Hypertext Preprocessor, język programowania, w którym napisany jest frontend Zabbixa i wiele innych aplikacji webowych · **docker** = narzędzie do kontenerów

TYLKO ODCZYT

Włączenie stref czasowych w bazie: bin/console database:enable_timezones

```
docker compose exec glpi bin/console database:enable_timezones
```

Obraz MariaDB ładuje dane stref czasowych przy pierwszym starcie, a ta komenda każe GLPI z nich korzystać. Potrzebne, gdy chcemy, żeby użytkownicy widzieli czasy w swojej strefie, a SLA liczyło się poprawnie po zmianie czasu.

□□ **GLPI** = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **SLA** = Service Level Agreement: umowa o poziomie usług, np. 99,9% dostępności · **docker** = narzędzie do kontenerów

□□ TYLKO ODCZYT

Sprawdzenie wersji obrazów: docker compose images

```
docker compose images
```

Przed każdą aktualizacją sprawdzamy, jakie wersje faktycznie działają, zamiast wierzyć plikowi `.env`.

□□ **docker** = narzędzie do kontenerów

□□ ZMIENIA SYSTEM

Kopia bazy GLPI: mariadb-dump

```
docker compose exec -T db sh -c 'mariadb-dump --single-transaction -u "$MARIADB_USER" -p"$MARIADB_PASSWORD" "$MARIADB_DATABASE"' > /docker/kopie/glpi/glpi-$(date +%F).sql
```

Zrzut logiczny bazy w spójnym stanie, bez zatrzymywania GLPI. Hasło czytamy ze zmiennych wewnątrz kontenera, więc nie trafia do historii powłoki. Katalog `/docker/kopie/glpi` zakładamy wcześniej (`sudo mkdir -p /docker/kopie/glpi`), tak jak katalog kopii MailCow.

□□ **GLPI** = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **docker** = narzędzie do kontenerów

□□ ZMIENIA SYSTEM

Kopia GLPI do repozytorium glpi: restic backup

```
export RESTIC_REPOSITORY=sftp:nazwa_usera@bkp01:/backup/glpi-restic
export RESTIC_PASSWORD_FILE=/docker/skrypty/restic-haslo-glpi.txt
docker compose -f /docker/compose/glpi/compose.yaml exec -T db \
  sh -c 'mariadb-dump --single-transaction -u "$MARIADB_USER" -p"$MARIADB_PASSWORD" "$MARIADB_DATABASE"' \
  > /docker/kopie/glpi/glpi.sql
restic backup /docker/kopie/glpi/glpi.sql /docker/apps/glpi/glpi /docker/apps/glpi/plugins /docker/compose/glpi --tag glpi
```

Rdzeń osobnego zadania `glpi`: świeży zrzut bazy, potem jedna migawka restic ze zrzutem, wolumenem `/var/glpi` (na hoście `/docker/apps/glpi/glpi`), katalogiem wtyczek wgranych ręcznie (`/docker/apps/glpi/plugins`) i plikiem Compose. Pełny skrypt z dziennikiem, retencją i timerem systemd o 03:00 opisuje strona Kopia całego docker01: /docker/apps, /docker/compose i bazy w kontenerach, na wzór zadania Vaultwardena.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **export** = ustaw zmienną środowiskową widoczną dla uruchamianych programów · **docker** = narzędzie do kontenerów · **sh** = shell: podstawowy interpreter poleceń Linuksa, tu uruchamia pobrany skrypt instalacyjny · **restic** = program kopii zapasowej z deduplikacją i szyfrowaniem, zapisujący repozytorium przez zwykłe konto SSH bez dodatkowego oprogramowania po drugiej stronie

TYLKO ODCZYT

Włączenie trybu konserwacji: bin/console maintenance:enable

```
docker compose exec glpi bin/console maintenance:enable
```

Użytkownicy widzą komunikat o konserwacji, a zgłoszenia mailem czekają w skrzynce. Włączamy przed aktualizacją i odtwarzaniem z kopii, a po pracy `bin/console maintenance:disable`.

docker = narzędzie do kontenerów

ZMIENIA SYSTEM

Aktualizacja GLPI: nowa wersja w .env i docker compose up -d

```
sudo sed -i 's/^GLPI_WERSJA=.*/GLPI_WERSJA=NowaWersja/' /docker/compose/glpi/.env
docker compose pull glpi
docker compose up -d glpi
```

Po starcie nowego obrazu skrypt startowy sam aktualizuje schemat bazy. To zmiana normalna, zwykle czerwona: aktualizacja schematu jest nieodwracalna bez odtworzenia z kopii, więc przed nią robimy zrzut bazy i kopię `/docker/apps/glpi`.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **sed** = stream editor: edytor strumienia tekstu · **docker** = narzędzie do kontenerów

TYLKO ODCZYT

Sprawdzenie integralności bazy po aktualizacji: bin/console database:check_schema_integrity

```
docker compose exec glpi bin/console database:check_schema_integrity
```

Porównuje strukturę bazy z tym, czego oczekuje zainstalowana wersja GLPI. Różnice zapisujemy w zgłoszeniu zmiany i sprawdzamy w dokumentacji, zanim wyłączymy tryb konserwacji.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01 · **docker** = narzędzie do kontenerów

KLIKAMY W INTERFEJSIE

Adres aplikacji: Setup → General → General setup

```
Setup → General → General setup → URL of the application: https://pomoc.zielona.example
→ Save
```

Od tego adresu zależą linki w powiadomieniach mailowych. Bez niego użytkownik dostaje w mailu link do wewnętrznej nazwy kontenera.

KLIKAMY W INTERFEJSIE

Logowanie kontem domenowym: Setup → Authentication → LDAP directories

```
Setup → Authentication → LDAP directories → Add → Server: dc01.ad.zielona.example, Port:
636
→ BaseDN: DC=ad,DC=zielona,DC=example → Login field: samaccountname → Test
```

Konto do odczytu katalogu bierzemy z jednostki kont technicznych, a drugi kontroler (`dc02`) dodajemy jako replikę. Dokładne pola i filtr na grupę opisujemy po instalacji w artykule bazy wiedzy.

LDAP = Lightweight Directory Access Protocol: protokół do odpytywania katalogu użytkowników (AD, FreeIPA)

KLIKAMY W INTERFEJSIE

Odbiór zgłoszeń mailem: Setup → Receivers

```
Setup → Receivers → Add → Name: it@zielona.example → Server: poczta.zielona.example
→ Connection options: IMAP, SSL, validate-cert, port 993 → Login i hasło skrzynki
→ Accepted mail archive folder: GLPI-przyjete → Refused mail archive folder: GLPI-
odrzucone → Save → Test
```

GLPI co kilka minut pobiera skrzynkę `it@zielona.example` z MailCow i zamienia każdy nowy mail w zgłoszenie, a odpowiedź z numerem zgłoszenia w temacie dopisuje do istniejącego.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01

KLIKAMY W INTERFEJSIE

Wysyłanie powiadomień: Setup → Notifications → Email notifications configuration

```
Setup → Notifications → Enable email notifications: Yes → Email notifications
configuration
→ Administrator email address: it@zielona.example → Way of sending emails: SMTP → Check
certificate: Yes
→ SMTP host: poczta.zielona.example, port 587, login i hasło konta wysyłającego → Save →
Send a test email to the administrator
```

Powiadomienia wychodzą z tego samego adresu, na który użytkownicy piszą, więc odpowiedź na powiadomienie trafia z powrotem do zgłoszenia.

KLIKAMY W INTERFEJSIE

Kalendarz godzin wsparcia: Setup → Dropdowns → Calendars

```
Setup → Dropdowns → Calendars → Add → Name: Godziny pracy IT
→ Time ranges: poniedziałek do piątku 07:00 do 16:00 → Close times: święta ustawowe
```

Kalendarz decyduje, kiedy zegar SLA biegnie. Drugi kalendarz „Całodobowy” zostawiamy pusty, bez przedziałów, dla SLA liczonego 24 godziny na dobę.

SLA = Service Level Agreement: umowa o poziomie usług, np. 99,9% dostępności

KLIKAMY W INTERFEJSIE

Poziomy usług: Setup → Service levels

```
Setup → Service levels → Add → Name: Zielona standard, Calendar: Godziny pracy IT → Save
→ zakładka SLA → Add: P1 podjęcie, Type: Time to own, Maximum time: 15 minutes
→ Add: P1 przywrócenie, Type: Time to resolve, Maximum time: 4 hours (i tak dla P2 do P4)
```

Każdy priorytet z tabeli na stronie o SLA dostaje dwa wpisy: czas podjęcia i czas rozwiązania. Poziomy eskalacji ustawiamy przy każdym wpisie, na przykład powiadomienie drugiego administratora 15 minut przed przekroczeniem.

SLA = Service Level Agreement: umowa o poziomie usług, np. 99,9% dostępności

KLIKAMY W INTERFEJSIE

Przypisanie SLA według priorytetu: Administration → Rules → Business rules for tickets

```
Administration → Rules → Business rules for tickets → Add → Criteria: Priority is Very high
→ Actions: Assign SLA TTO: P1 podjęcie, Assign SLA TTR: P1 przywrócenie → Active: Yes
```

Jedna reguła na jeden priorytet. Gdy SLA ustawi ręcznie użytkownik albo szablon zgłoszenia, reguła go nie nadpisze.

SLA = Service Level Agreement: umowa o poziomie usług, np. 99,9% dostępności

KLIKAMY W INTERFEJSIE

Macierz priorytetu: Setup → General → Assistance

```
Setup → General → Assistance → Matrix of calculus for priority → zostawiamy poziomy Low, Medium, High
dla Impact i Urgency → wypełniamy macierz według tabeli ze strony o SLA → Save
```

GLPI ma pięć poziomów wpływu i pilności i sześć priorytetów; my używamy trzech i trzech. Przyjmujemy przełożenie: P1 to Very high, P2 to High, P3 to Medium, P4 to Low.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01

KLIKAMY W INTERFEJSIE

Kategorie zgłoszeń: Setup → Dropdowns → ITIL categories

```
Setup → Dropdowns → ITIL categories → Add → Name: Optima → Visible for an incident: Yes
→ Visible for a request: Yes → Group in charge: IT L1 → Template for an incident:
Incydent
```

Kategorie odpowiadają usługom z katalogu: Optima, WMS, Poczta, Sieć i VPN, Konta i dostępy, Sprzęt, Drukowanie, Aplikacje wewnętrzne, Serwery, Bezpieczeństwo. Od kategorii zależy domyślna grupa, czyli kolejka, i szablon zgłoszenia.

ITIL = Information Technology Infrastructure Library: zbiór dobrych praktyk zarządzania usługami IT, dziś rozwijany przez PeopleCert; wersja 4 od 2019, wersja 5 od 2026 · **WMS** = Warehouse Management System: system zarządzania magazynem · **VPN** = Virtual Private Network: szyfrowany tunel do sieci firmowej

KLIKAMY W INTERFEJSIE

Szablony zgłoszeń: Setup → Dropdowns → Ticket templates

```
Setup → Dropdowns → Ticket templates → Add → Name: Incydent
→ zakładka Mandatory fields: Title, Description, Category, Location
→ zakładka Predefined fields: Type: Incident
```

Drugi szablon „Wniosek” ma typ Request i pole zatwierdzającego. Pola obowiązkowe odpowiadają minimum danych z Fiszki na stronie o incydentach.

KLIKAMY W INTERFEJSIE

Kolejki jako grupy: Administration → Groups

```
Administration → Groups → Add → Name: IT L1 → Can be in charge: Yes
→ tak samo: IT L2, Zespół aplikacji (Ola i jej zespół, profil Technician)
```

Kolejką w GLPI jest grupa, do której przypisujemy zgłoszenie. Dostawców prowadzimy osobno, w Management → Suppliers, i przypisujemy do zgłoszenia jako dostawcę.

GLPI = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01

□□ KLIKAMY W INTERFEJSIE

Katalog usług: Administration → Forms

```
Administration → Forms → Add → Name: Dostęp czasowy do bazy danych (KU-05)
→ pytania: baza, zakres (odczyt albo zapis), data końca, uzasadnienie
→ Destinations: Ticket (Type: Request), Approval: właściciel danych → Access: Private,
grupa Programiści
```

Formularze z GLPI 11 i nowszych zastępują dawną wtyczkę Formcreator i pokazują się w Assistance → Service catalog jako kafelki. Kategorie katalogu ustawiamy w Setup → Dropdowns → Service catalog.

□□ **GLPI** = Gestionnaire Libre de Parc Informatique: otwarty system do zgłoszeń, inwentarza i bazy wiedzy IT; w Zielonej działa w kontenerze na docker01

Z życia wzięte

□□ Z ŻYCIA WZIĘTE

Środa, 8:00, pierwszy tydzień z GLPI. Pani Halina, zamiast dzwonić na komórkę, pisze na `it@zielona.example`, że nie widzi udziału z fakturami, i po minucie dostaje odpowiedź z numerem zgłoszenia i nazwiskiem dyżurnego. Marek zgłasza przez portal awarię skanera i od razu wybiera pilność, a GLPI sam liczy P3 i ustawia czas. W piątek pan Zbigniew dostaje od Tomka pierwszy raport: 47 zgłoszeń, z tego 31 wniosków, 14 incydentów, 2 zmiany, dwa przekroczone czasy, oba przez czekanie na dostawcę.

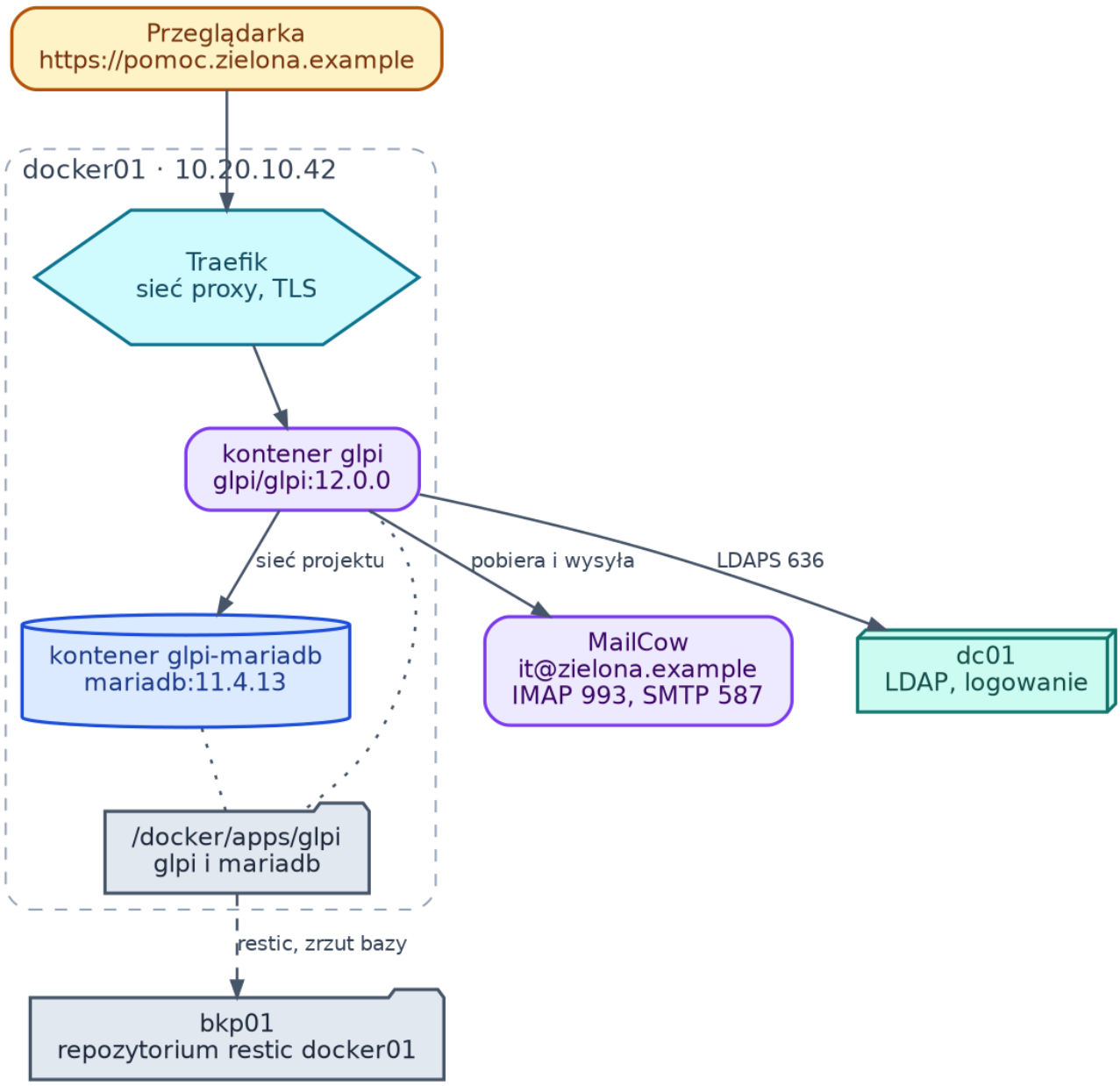
Jak to działa

□□ Rozwiń: jak to działa pod spodem

Pojęcie: GLPI. Wolne oprogramowanie do obsługi zgłoszeń i ewidencji sprzętu, rozwijane od 2003 roku, najpierw przez francuskie stowarzyszenie INDEPNET, a od 2015 roku pod kierunkiem firmy Teclib'. Nazwa pochodzi od francuskiego Gestionnaire Libre de Parc Informatique, czyli „wolny menedżer parku komputerowego”. Licencja to GPL w

wersji 3, bez opłat za użytkowników.

GLPI to aplikacja PHP z bazą MariaDB albo MySQL. W Zielonej oba składniki działają jako kontenery na `docker01`, ruch z przeglądarek przechodzi przez Traefika, poczta przychodzi i wychodzi przez MailCow, a logowanie sprawdza kontroler domeny.



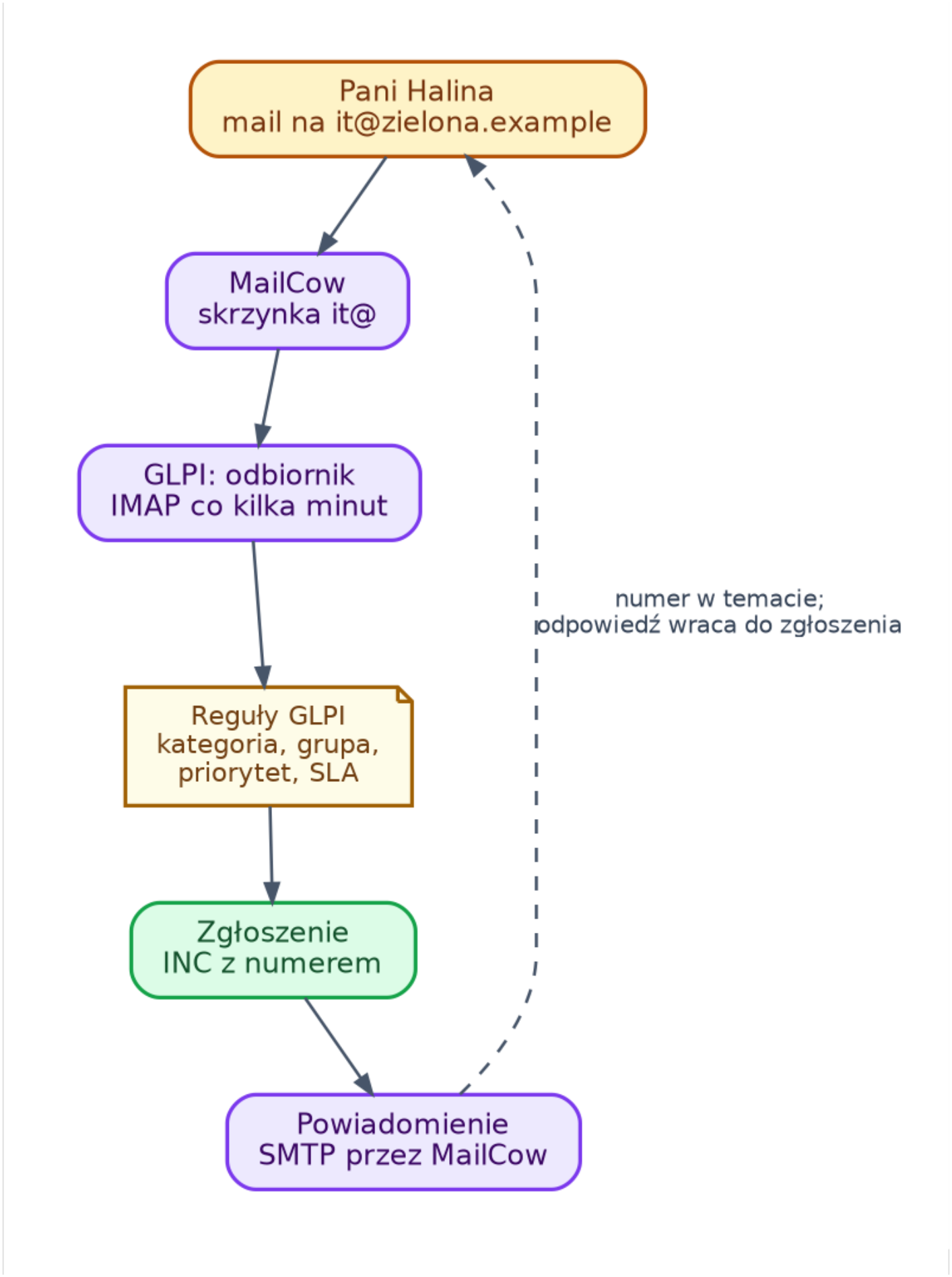
Rysunek 1. GLPI na docker01: przeglądarka przez Traefika, kontener GLPI z danymi w /docker/apps/glpi/glpi, kontener MariaDB w sieci projektu, MailCow do poczty, dc01 do logowania i bkp01 jako miejsce kopii (kliknięcie otwiera pełny rozmiar)

Dlaczego oficjalny obraz. Projekt GLPI publikuje własny obraz `glpi/glpi` na Docker Hub, z wersjami przypiętymi do wydania (`12.0.0`, `11.0.11`) i obrazami odświeżanymi dla każdej wspieranej gałęzi. Obraz działa jako użytkownik bez uprawnień administratora, ma wbudowane zadania cykliczne GLPI (uruchamiane co minutę) i sam instaluje albo aktualizuje bazę, gdy dostanie dane połączenia. Obrazy społeczności, popularne kilka lat temu, nie są nam potrzebne.

Dlaczego wersja 12. GLPI 12.0.0 ukazał się 7 października 2026 roku, a jego ukazanie się zamyka wsparcie GLPI 10. Gałąź 11 nadal dostaje poprawki bezpieczeństwa (ostatnia 11.0.11 z 1 października 2026). Dla nowej instalacji bez danych do migracji wybieramy 12, żeby za rok nie robić dużej aktualizacji; ceną jest to, że to wydanie ma kilka dni i pierwsze poprawki dopiero przyjdą. GLPI 12 wymaga PHP 8.3 (jest w obrazie) i co najmniej MariaDB 10.11; MariaDB 11.4 to wydanie z długim wsparciem, a GLPI sprawdza tylko wersję minimalną.

Droga zgłoszenia mailem

Większość zgłoszeń w małej firmie przychodzi mailem, więc ta droga musi działać bez wyjątków. Mail od użytkownika trafia do skrzynki w MailCow, GLPI go pobiera, zakłada zgłoszenie, reguły nadają kategorię i grupę, a powiadomienie z numerem wraca do użytkownika. Odpowiedź użytkownika z numerem w temacie trafia do tego samego zgłoszenia.


Pani Halina
mail na it@zielona.example
MailCow
skrzynka it@
GLPI: odbiornik
IMAP co kilka minut
Reguły GLPI
kategoria, grupa,
priorytet, SLA
Zgłoszenie
INC z numerem
Powiadomienie
SMTP przez MailCow
numer w temacie;
odpowiedź wraca do zgłoszenia

Rysunek 2. Zgłoszenie mailem: użytkownik pisze na it@zielona.example, MailCow trzyma wiadomość, GLPI pobiera ją przez IMAP, reguły nadają kategorię, priorytet i SLA, powiadomienie z numerem wraca przez SMTP (kliknięcie otwiera pełny rozmiar)

Jak standardy z tej książki mapują się na GLPI

Standard z tej książki	Obiekt w GLPI	Gdzie
Rodzaje pracy: incydent, wniosek	typ zgłoszenia Incident i Request	Assistance → Tickets
Problem i znany błąd	Problem z zakładką Analysis (przyczyny, objawy)	Assistance → Problems
Zmiana i wniosek o zmianę	Change z zakładkami Analysis i Plans (plan wdrożenia, plan wycofania, lista kontrolna)	Assistance → Changes
Katalog usług	formularze z miejscem docelowym Ticket	Administration → Forms
Kolejki L1, L2, zespół Oli	grupy techniczne	Administration → Groups
Dostawcy i ich umowy (UC)	dostawcy i kontrakty	Management → Suppliers, Contracts
SLA i OLA	poziomy usług z wpisami SLA i OLA; GLPI 12 pozwala przypisać OLA do grupy	Setup → Service levels
Godziny wsparcia	kalendarz	Setup → Dropdowns → Calendars
Macierz priorytetu	macierz wpływ i pilność	Setup → General → Assistance
Baza wiedzy dla użytkowników	krótkie odpowiedzi z linkiem do BookStacka	Tools → Knowledge base
Sprzęt użytkowników	komputery przypisane do użytkownika	Assets → Computers

Elementów konfiguracji serwerowni nie dublujemy w GLPI: źródłem prawdy jest NetBox, a w zgłoszeniu wpisujemy nazwę elementu taką jak w NetBoksie. Czy łączyć oba narzędzia wtyczką albo przez API, decydujemy po kilku miesiącach pracy.

Pierwsze kroki po instalacji

1. Logujemy się kontem `glpi` z hasłem podanym w powitaniu w dzienniku kontenera i od razu je zmieniamy.

2. Domyślne konta `tech`, `normal` i `post-only` (ich hasła są w dokumentacji GLPI) wyłączamy albo zmieniamy im hasła; GLPI pokazuje ostrzeżenie, dopóki tego nie zrobimy.
3. Ustawiamy adres aplikacji, logowanie z domeny, odbiornik poczty i powiadomienia według Fiszki.
4. Zakładamy kalendarz, poziomy usług, reguły SLA, macierz priorytetu, grupy, kategorie i szablony.
5. Budujemy formularze katalogu usług według tabeli ze strony o wnioskach.
6. Przez tydzień pilotażu zgłoszenia zakłada tylko zespół IT i dwóch użytkowników (pani Halina i Marek), potem ogłaszamy adres całej firmie.

> ◫ Zrzut do uzupełnienia: ekran Setup → Service levels → „Zielona standard”, zakładka SLA z ośmioma wpisami (P1 do P4, podjęcie i przywrócenie), kolumny Type i Maximum time widoczne, kalendarz „Godziny pracy IT” w nagłówku.

> ◫ Zrzut do uzupełnienia: Assistance → Service catalog widziany kontem zwykłego użytkownika, kafelki KU-01 do KU-11 z opisami, kafelek „Dostęp czasowy do bazy danych” widoczny tylko dla grupy programistów.

> ◫ Zrzut do uzupełnienia: zgłoszenie INC z pierwszego tygodnia, prawy panel z priorytetem P1 (Very high), polami Time to own i Time to resolve z odliczaniem oraz przypisaną grupą IT L1.

Kopia zapasowa

GLPI ma własne repozytorium restic `glpi` z poziomem 5 lat, decyzją autora z 11 października 2026. Zgłoszenia, zmiany i wnioski o odejściu pracowników są dowodem dla audytora, a poziom dotyczy całego repozytorium, więc GLPI nie może zostać w repozytorium `docker01` z poziomem 90 dni. Drabinę retencji opisuje strona Strategia kopii zapasowych Zielonej: RPO, RTO, reguła 3-2-1-1-0 i dwie lokalizacje; dla GLPI oznacza to:

Szczebel	Gdzie	Ile trzymamy
kopie dzienne	`bkp01`, repozytorium `glpi`	7 dni

Szczebel	Gdzie	Ile trzymamy
kopie tygodniowe	`bkp01`, repozytorium `glpi`	90 dni
kopia krzyżowa	`bkp02`	30 dni
paczki miesięczne	zimne archiwum w chmurze	180 dni
paczki kwartalne	zimne archiwum w chmurze	365 dni
paczka roczna	zimne archiwum z niezmiennością	5 lat

Do kopii wchodzą trzy rzeczy: zrzut bazy MariaDB zrobiony `mariadb-dump` tuż przed migawką, wolumen `/var/glpi` (na hoście `/docker/apps/glpi/glpi`) z konfiguracją, kluczem szyfrowania, załącznikami, wtyczkami z Marketplace i logami oraz katalog `/docker/apps/glpi/plugins` z wtyczkami wgranymi ręcznie. Plików bazy z `/docker/apps/glpi/mariadb` nie kopiujemy, bo kopiowane w trakcie zapisu dają niespójny stan. Zadanie `glpi` działa osobno od kopii `docker01`, z własnym hasłem repozytorium i timerem systemd o 03:00, a skrypt `docker01` pomija `/docker/apps/glpi` w całości; szczegóły i skrypt są na stronie Kopia całego docker01: /docker/apps, /docker/compose i bazy w kontenerach. Hasło repozytorium `glpi` trzymamy w Vaultwardenie i, jak pozostałe hasła repozytoriów, na wydruku w sejfie.

W samym GLPI wyłączamy automatyczne czyszczenie zamkniętych zgłoszeń, żeby bieżąca baza trzymała pełną historię, a kopie chroniły ją przed awarią i błędnym usunięciem. W GLPI 12 ustawienia przechowywania danych są w osobnej zakładce Data and Privacy.

> ⚠ Katalog `/docker/apps/glpi/glpi/config` zawiera klucz, którym GLPI szyfruje hasła zapisane w bazie (na przykład hasło skrzynki `it@zielona.example`). Zrzut bazy bez tego katalogu da się odtworzyć, ale zapisane hasła trzeba będzie wpisać od nowa.

Aktualizacja

Aktualizacja poprawkowa w ramach tej samej wersji głównej (na przykład 12.0.0 do 12.0.1) może trafić na listę zmian standardowych, gdy zrobimy ją kilka razy bez kłopotów. Aktualizacja do nowej wersji głównej jest zawsze zmianą normalną, czerwoną. Kolejność: przeczytanie informacji o wydaniu, tryb konserwacji, ręczne uruchomienie zadania kopii `glpi`, zmiana wersji w `.env`, `pull` i `up -d`, dziennik do powitania, sprawdzenie integralności, test zgłoszenia mailem, wyłączenie trybu konserwacji. Na laboratorium najpierw odtwarzamy wczorajszą kopię i próbujemy tam.

Porównanie z innymi narzędziami

Licencje podajemy tylko jako informację; cen nie podajemy, bo zmieniają się częściej niż ta strona.

	GLPI	Jira Service Management	Zammad	iTop
Kto rozwija	projekt GLPI, Teclib'	Atlassian	Zammad GmbH i społeczność	Combodo
Licencja	GPL 3, bez opłat	komercyjna; Data Center przestaje być sprzedawany nowym klientom po 30 marca 2026, koniec wsparcia 28 marca 2029, zostaje chmura	AGPL 3, bez opłat licencyjnych	AGPL 3
Własny serwer	tak, oficjalny obraz Dockera	po 2029 tylko chmura	tak	tak
Mocna strona	zgłoszenia, ewidencja sprzętu, SLA i OLA, katalog usług w jednym	integracja z Jira Software, dojrzałe procesy ITSM, automatyzacje	wygodna skrzynka zgłoszeń i czat, prosty interfejs	rozbudowana CMDB i analiza wpływu, ITIL od podstaw
Słabsza strona	interfejs bywa gęsty, dużo opcji	koszt i zależność od chmury jednego dostawcy	mniej funkcji ITIL (zmiany, problemy, CMDB)	dłuższe wdrożenie, mniejsza społeczność w Polsce
Dla Zielonej	wybór: darmowy, na naszym serwerze, wystarcza na lata	sensowny, gdyby Zielona już używała Jira	sensowny, gdyby chodziło tylko o skrzynkę zgłoszeń	sensowny, gdyby CMDB była ważniejsza niż NetBox

Częste błędy

Objaw	Przyczyna	Co robimy
Kontener `glpi` kończy pracę z komunikatem, że katalog nie jest zapisywalny	katalog `/docker/apps/glpi/glpi` należy do roota	`chown` na UID użytkownika z obrazu, według Fiszki
Kontener GLPI czeka w nieskończoność na bazę	baza nie przeszła testu zdrowia albo złe hasło w `.env`	`docker compose logs db`, porównanie zmiennych w `.env`

Objaw	Przyczyna	Co robimy
Linki w mailach prowadzą do `http://glpi` albo do adresu IP	nieustawiony adres aplikacji	Setup → General → URL of the application
Maile do `it@` nie tworzą zgłoszeń	odbiornik z błędem połączenia albo wyłączone zadanie pobierania poczty	przycisk Test w odbiorniku, lista zadań automatycznych w Setup → Automatic actions
Odpowiedź użytkownika tworzy nowe zgłoszenie zamiast dopisać się do starego	użytkownik zmienił temat i usunął numer zgłoszenia	prosimy o odpowiadanie na powiadomienie bez zmiany tematu, zgłoszenia łączymy ręcznie
SLA liczy czas w nocy	poziom usług bez kalendarza	kalendarz „Godziny pracy IT” przy poziomie usług
Każde zgłoszenie ma priorytet Very high	użytkownicy wybierają pilność bez zrozumienia	w portalu tylko pilność z opisem słownym, wpływ ustala dyżurny
Ostrzeżenie o domyślnych hasłach nie znika	konta `tech`, `normal`, `post-only` z domyślnymi hasłami	zmiana haseł albo wyłączenie kont
Po aktualizacji obrazu wtyczka zniknęła z listy	wtyczka wgrana ręcznie do `/var/www/glpi/plugins` bez montowania, więc odtworzony kontener ma pusty katalog	wtyczki z Marketplace (lądują w `/var/glpi/marketplace`), ręczne tylko do zamontowanego `/docker/apps/glpi/plugins`
Po aktualizacji błąd bazy	przerwana aktualizacja schematu	tryb konserwacji, `database:check_schema_integrity`, w razie potrzeby odtworzenie zrzutu sprzed aktualizacji

□□ Ćwiczenia

1. Po `docker compose up -d` kontener `glpi` restartuje się w kółko, a w dzienniku jest komunikat o katalogu niezapisywalnym przez UID 33. Co robimy?

Odpowiedź

`sudo chown -R 33:33 /docker/apps/glpi/glpi` (UID z komunikatu albo z `docker run --rm --entrypoint id`), potem `docker compose up -d` i obserwacja dziennika do powitania.

1. Pani Halina odpowiada na powiadomienie z GLPI, ale zmienia temat na „Pilne!!!”. Co się stanie i jak temu zapobiec?

Odpowiedź

GLPI nie znajdzie numeru zgłoszenia w temacie i utworzy nowe zgłoszenie. Łączymy je ręcznie z pierwotnym, a w instrukcji dla użytkowników piszemy, żeby odpowiadać bez zmiany tematu; pilność zmienia się w portalu albo telefonem do dyżurnego.

1. Planujemy aktualizację GLPI z 12.0.0 do 12.1.0. Jaki to rodzaj zmiany i jakie są pierwsze trzy kroki?

Odpowiedź

Zmiana normalna, czerwona, bo aktualizacja zmienia schemat bazy. Pierwsze kroki: przeczytanie informacji o wydaniu (wymagania, zmiany niekompatybilne), próba na laboratorium z odtworzonej kopii, a w oknie: tryb konserwacji i zrzut bazy z kopią katalogów `/docker/apps/glpi/glpi` i `/docker/apps/glpi/plugins`.

1. Dlaczego nie publikujemy portu 80 kontenera GLPI na hoście, skoro „tak byłoby prościej”?

Odpowiedź

Bo na `docker01` porty 80 i 443 należą do Traefika, który kończy TLS i kieruje ruch po nazwie hosta. Opublikowany port ominąłby szyfrowanie i dałby drugą, niekontrolowaną drogę do aplikacji. GLPI jest dostępny tylko przez `pomoc.zielona.example`.

Powiązane: Jak trzymamy aplikacje w Dockerze: konwencje Zielonej · Reverse proxy w Dockerze: Traefik przed wszystkimi aplikacjami na docker01 · MailCow: codzienna praca z domenami, skrzynkami i spamem · Service desk, linie wsparcia i SLA: kto odbiera zgłoszenie i w jakim czasie · Wnioski o usługę, katalog usług i onboarding: od nowego konta do odebrania dostępu

Źródła: github.com/glpi-project/glpi, wydania 12.0.0 (7 października 2026) i 11.0.11 (1 października 2026), licencja GPL 3 · glpi-project.org, „What's New in GLPI 12” (wymagania PHP 8.3, MariaDB 10.11 albo MySQL 8.0) i „GLPI 10 end of support is coming soon” (5 października 2026) · github.com/glpi-project/glpi, kod 12.0.0: polecenia `system:check_requirements`, `maintenance:enable`, `database:check_schema_integrity`, `database:enable_timezones`, sprawdzenie minimalnej wersji bazy · github.com/glpi-project/docker-images, Dockerfile, README, `docker-compose.example.yml` i skrypty startowe obrazu `glpi/glpi` (wolumen `/var/glpi` z `config`, `files`, `logs` i `marketplace`, `GLPI_MARKETPLACE_DIR=/var/glpi/marketplace` poza gałęzią 10, osobne montowanie `/var/www/glpi/plugins` dla własnych wtyczek, stan 19 października 2026; repozytorium obrazu nie publikuje wydań na GitHubie, tagi są tylko na Docker Hub; użytkownik `www-data`, automatyczna instalacja i aktualizacja, powitanie z kontem `glpi`) · hub.docker.com/r/glpi/glpi, opis zmiennych i tagi (12.0.0, 11.0.11, stan 11 października 2026; tag 12.0.0 potwierdzony ponownie 19 października 2026) · hub.docker.com, obraz `mariadb`, tag 11.4.13 · github.com/MariaDB/mariadb-docker, `docker-entrypoint.sh` (ładowanie stref czasowych) i `healthcheck.sh` · help.glpi-project.org: „Forms” (katalog usług od GLPI 11), „Service Levels” (SLA, OLA, kalendarz, eskalacje, reguły),

„Receivers” (odbiór poczty) · neteye-blog.com, „GLPI 12: A Practical Review of the Major Changes” (30 września 2026; OLA przypisane do grupy, zakładka Data and Privacy) · en.wikipedia.org/wiki/GLPI (historia, INDEPNET, Teclib') · Atlassian Data Center: jimiwikman.se, „Atlassian Data Center is officially ending on March 28, 2029” (ogłoszenie z 8 września 2025) · support.zammad.com, „Is there a license fee for using Zammad?” (AGPL 3) · github.com/Combodo/iTop (AGPL 3).

Revision #7
Created 2026-10-11 14:27:02 UTC by Tomasz Konieczny
Updated 2026-10-26 14:25:42 UTC by Tomasz Konieczny