Portainer: Docker w przeglądarce

Typ: procedura **Dotyczy: Portainer CE 2.45.1** **Czas: ~30 min**

Efekt: panel WWW pod `https://portainer.zielona.example`, w którym widzimy stan wszystkich kontenerów na `docker01` jednym spojrzeniem, bez wpisywania `docker compose ps` w każdym z kilkunastu katalogów osobno.

Skróty użyte na tej stronie (6)

Skrót	Co znaczy
CE	Community Edition: darmowa, społecznościowa wersja Dockera, w odróżnieniu od płatnej wersji Enterprise
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
LDAP	Lightweight Directory Access Protocol: protokół do odpytywania katalogu użytkowników (AD, FreeIPA)
LTS	Long Term Support: wersja z długim wsparciem poprawkami
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS

Zanim zaczniemy

> **Z ŻYCIA WZIĘTE**
>
> Czwartek, 10:15. Tomek szuka, dlaczego kontener NetBoksa nie wstał po restarcie hosta. Zamiast wchodzić po SSH i wpisywać `cd /docker/compose/...` po kolei w ośmiu katalogach, otwiera Portainera, widzi na liście, że akurat ten jeden kontener stoi w stanie `Exited`, i wchodzi od razu w jego logi jednym kliknięciem.

Procedura zakłada działający Docker Engine z wtyczką Compose, zainstalowany według strony Instalacja Dockera na Rocky Linux 9 z osobnym wolumenem /docker, katalogi zgodne z konwencją opisaną na stronie

Jak trzymamy aplikacje w Dockerze: konwencje Zielonej, i działającego Traefika przed aplikacjami, opisanego na stronie Reverse proxy w Dockerze: Traefik przed wszystkimi aplikacjami na docker01. Komendy wykonujemy z CLI bash jako user z uprawnieniami sudo albo jako user należący do grupy `docker`.

> 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.

Wersja obrazu `portainer/portainer-ce:2.45.1` sprawdzona w wydaniach projektu na GitHubie 28 września 2026, gałąź LTS. Przy kolejnej instalacji sprawdzamy aktualną wersję, zanim wpiszemy ją do pliku Compose.

Przebieg

1. Założenie katalogów

```
sudo mkdir -p /docker/apps/portainer /docker/compose/portainer
```

Portainer trzyma własne dane (konta, ustawienia, historię) w jednym katalogu, bez osobnej bazy danych na zewnątrz.

2. Plik compose.yaml

```
sudo vim /docker/compose/portainer/compose.yaml
```

```
services:
  portainer:
    image: portainer/portainer-ce:2.45.1
    container_name: portainer
    restart: unless-stopped
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /docker/apps/portainer/data:/data
    networks:
      - proxy
```

```
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.portainer.rule=Host(`portainer.zielona.example`)"
      - "traefik.http.routers.portainer.entrypoints=websecure"
      - "traefik.http.routers.portainer.tls=true"
      - "traefik.http.services.portainer.loadbalancer.server.port=9000"

networks:
  proxy:
    external: true
```

> Gniazdo `/var/run/docker.sock` w tym kontenerze montujemy **bez** `:ro`, w odróżnieniu od Traefika na stronie Reverse proxy w Dockerze. Portainer nie tylko czyta stan kontenerów, ale i zarządza nimi: startuje, zatrzymuje, usuwa, tworzy nowe. Dostęp do tego gniazda, w dowolnym trybie, jest równoznaczny z pełnymi prawami roota na `docker01`, opisanymi w ostrzeżeniu na stronie Instalacja Dockera na Rocky Linux 9 przy grupie `docker`. Panel Portainera zabezpieczamy więc kontem administratora z mocnym hasłem od pierwszej minuty, nie zostawiamy go dostępnego bez logowania nawet na chwilę.

3. Pierwsze uruchomienie

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

Przykład:

```
[+] Running 2/2
 ✔ Network proxy         Created
 ✔ Container portainer   Started
```

4. Założenie konta administratora

```
docker compose logs portainer --tail=5
```

Otwieramy `https://portainer.zielona.example` w przeglądarce i zakładamy pierwsze konto administratora.

>

> Portainer daje na to**limitowany czas od pierwszego uruchomienia kontenera**, zwykle kilka minut: jeśli nikt nie założy konta administratora w tym oknie, panel przechodzi w tryb, w którym każdy, kto zdąży wejść pierwszy, może założyć konto administratora zamiast nas. Zakładamy konto od razu po `docker compose up -d`, nie zostawiamy tego na później.

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ekran „Create the initial administrator user” w Portainerze, z widocznym polem hasła i licznikiem pozostałego czasu w nagłówku.

Weryfikacja

```
docker compose ps
```

Przykład:

```
NAME        IMAGE                          STATUS
portainer   portainer/portainer-ce:2.45.1  Up 2 minutes
```

```
curl -Ik https://portainer.zielona.example
```

Przykład:

```
HTTP/2 200
```

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: widok „Environments” po zalogowaniu, z lokalnym środowiskiem `docker01` i liczbą uruchomionych kontenerów widoczną na kafelku.

Wycofanie zmian

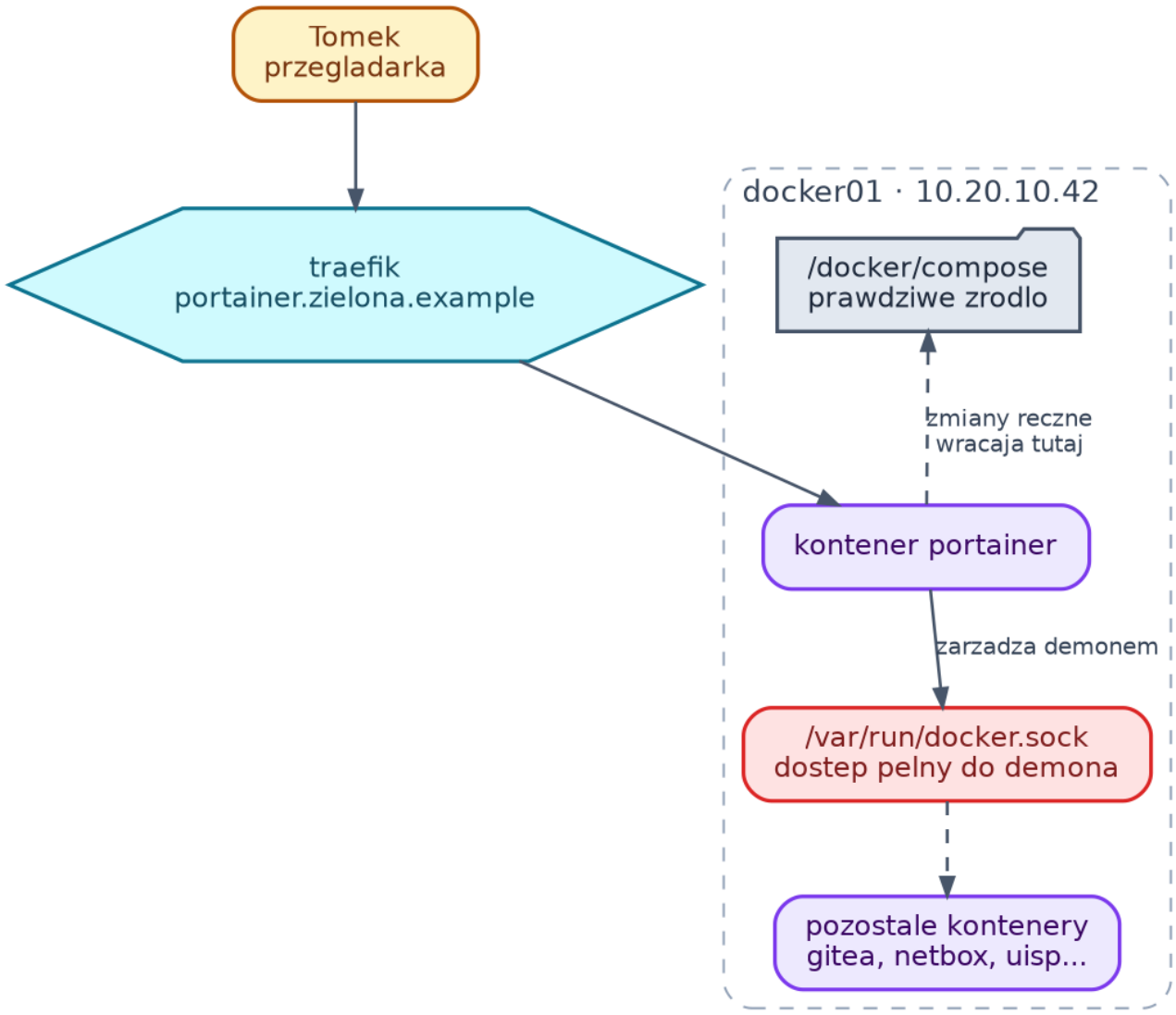
```
cd /docker/compose/portainer
docker compose down
```

Zatrzymuje i usuwa kontener Portainera bez wpływu na aplikacje, którymi zarządzał: same kontenery dalej działają, tylko bez panelu do ich podglądu.

Jak to działa

Rozwiń: jak to działa pod spodem

> **Pojęcie: Portainer.** Panel WWW do zarządzania Dockerem przez przeglądarkę zamiast wyłącznie przez CLI: lista kontenerów, obrazów, sieci i wolumenów, podgląd logów, konsola do kontenera, restart i usuwanie, jednym kliknięciem zamiast osobnej komendy na każdą z tych czynności.



Portainer na docker01 zarządza demonem Dockera przez docker.sock, a prawdziwym źródłem konfiguracji zostaje /docker/compose (kliknięcie otwiera pełny rozmiar)

LDAP w Portainer CE

Portainer Community Edition, w odróżnieniu od bezpłatnych, ale ograniczonych wersji wielu podobnych narzędzi, ma logowanie przez LDAP wbudowane bez dopłaty: sekcja Settings → Authentication → LDAP pozwala wskazać kontroler domeny, konto do wyszukiwania i wzorzec grupy, dokładnie tym samym mechanizmem co logowanie LDAP opisane dla Gitea na stronie Logowanie przez LDAP. Konfigurację krok po kroku odkładamy do osobnej strony integracji z domeną, tu odnotowujemy tylko, że funkcja jest dostępna w wersji, z której korzysta Zielona.

Co robimy w Portainerze, a czego nie

Portainer daje możliwość edycji `compose.yaml` wprost w panelu i uruchomienia nowego stosu przyciskiem, ale w Zielonej tego nie robimy. Jedynym prawdziwym źródłem konfiguracji każdej aplikacji zostaje plik pod `/docker/compose/<nazwa>`, opisany na stronie Jak trzymamy aplikacje w Dockerze: konwencje Zielonej: trafia do repozytorium konfiguracji, ma historię zmian i kopię zapasową.

W Portainerze robimy	W Portainerze nie robimy
podgląd stanu, logów i zużycia zasobów kontenerów	tworzenie nowego stosu przyciskiem „Add stack”
szybki restart pojedynczego kontenera w sytuacji awaryjnej	trwałą zmianę zmiennych środowiskowych przez edytor w panelu
konsolę do kontenera do doraźnej diagnostyki	zmianę wolumenów albo portów istniejącej usługi w panelu
przegląd, które obrazy i wolumeny zalegają bez użycia	`docker system prune` bez wcześniejszego sprawdzenia z `docker system df -v`, patrz Utrzymanie hosta dockerowego

Zmiana wprowadzona ręcznie w panelu, z pominięciem pliku Compose, znika albo zostaje nadpisana przy najbliższym `docker compose up -d` wykonanym z linii komend, bo Compose porównuje stan kontenera z plikiem na dysku, nie z tym, co ktoś kiedyś kliknął w przeglądarce. Traktujemy Portainera jako okno podglądowe i narzędzie awaryjne, nie jako miejsce, gdzie żyje konfiguracja.

Częste błędy

Objaw	Przyczyna	Co robimy
Panel po otwarciu każe zakładać konto administratora na nowo	nikt nie zdążył założyć konta w limitowanym oknie czasowym po starcie	`docker compose restart portainer`, zakładamy konto od razu przy nowym oknie
Kontener zatrzymany w Portainerze wstaje z powrotem po chwili	Compose z linii komend widzi politykę `unless-stopped` i wraca do zgodności z plikiem przy najbliższym `up -d` z crona albo innej automatyzacji	zatrzymanie na stałe robimy zmianą w `compose.yaml`, nie przyciskiem w panelu
Zmiana zrobiona w panelu znika po następnym wdrożeniu z linii komend	konfiguracja żyje w pliku Compose, panel tylko ją pokazuje	zmianę wprowadzamy w pliku pod `/docker/compose`, potem `docker compose up -d`

Objaw	Przyczyna	Co robimy
Ktoś spoza zespołu IT ma pełny dostęp do wszystkich kontenerów	konto w Portainerze założone bez ograniczenia ról albo hasło udostępnione dalej	ograniczamy konta do zespołu IT, rotujemy hasło administratora

Ćwiczenia

1. Zakładamy Portainera na `docker01`, podłączonego do sieci `proxy`, i zakładamy konto administratora w oknie czasowym po pierwszym starcie.
2. Wyjaśniamy, dlaczego zmiana zmiennej środowiskowej zrobiona w panelu Portainera dla istniejącego kontenera nie jest bezpiecznym miejscem do trzymania konfiguracji na stałe.

Odpowiedź

Bo prawdziwym źródłem konfiguracji w Zielonej jest plik `compose.yaml` pod `/docker/compose/<nazwa>`. Zmiana zrobiona wyłącznie w panelu nie trafia do tego pliku, więc znika albo zostaje nadpisana przy kolejnym `docker compose up -d` wykonanym z linii komend na podstawie pliku, a poza tym nie ma żadnej historii ani kopii zapasowej, w odróżnieniu od pliku w repozytorium konfiguracji.

1. Dlaczego montowanie `docker.sock` do kontenera Portainera jest równoznaczne z nadaniem mu pełnych praw roota na hoście, mimo że sam kontener Portainera działa jako zwykły proces?

Odpowiedź

Bo demon Dockera, z którym Portainer rozmawia przez to gniazdo, sam działa jako root i wykonuje dla dowolnego klienta gniazda dowolne polecenie, łącznie z uruchomieniem nowego kontenera z zamontowanym dowolnym katalogiem hosta. Ktokolwiek ma dostęp do gniazda, może przez nowy kontener odczytać albo zmienić cokolwiek na hoście, niezależnie od uprawnień konta systemowego, które uruchomiło sam kontener Portainera.

Powiązane: Jak trzymamy aplikacje w Dockerze: konwencje Zielonej · Reverse proxy w Dockerze: Traefik przed wszystkimi aplikacjami na docker01 · Instalacja Dockera na Rocky Linux 9 z osobnym wolumenem /docker · Logowanie przez LDAP

Źródła: docs.portainer.io, rozdział „Install Portainer CE with Docker on Linux" · docs.portainer.io, rozdział „Authenticate via LDAP" · github.com/portainer/portainer, wydania (releases)

Revision #22
Created 2026-09-28 17:15:46 UTC by Tomasz Konieczny

Updated 2026-10-26 14:09:34 UTC by Tomasz Konieczny