



Ernabo Adrian Flak

★★★★★ 4,6 / 5

745 ocen

Szkolenie: Docker i Konteneryzacja dla aplikacji PHP

Numer usługi 2025/08/18/22948/2945443

📍 zdalna w czasie rzeczywistym

🏠 Usługa szkoleniowa

🕒 13 h

📅 11.03.2026 do 14.03.2026

1 560,00 PLN brutto

1 560,00 PLN netto

120,00 PLN brutto/h

120,00 PLN netto/h

Informacje podstawowe

Kategoria	Informatyka i telekomunikacja / Programowanie
Identyfikatory projektów	Małopolski Pociąg do kariery, Kierunek - Rozwój
Grupa docelowa usługi	Programiści i developerzy PHP (junior/mid), którzy chcą usprawnić proces uruchamiania i utrzymywania aplikacji. Administratorzy systemów i DevOps, którzy potrzebują zrozumieć podstawy Dockera w kontekście aplikacji PHP. Osoby z podstawową znajomością Linuxa i PHP, które chcą nauczyć się konteneryzacji aplikacji webowych. Zespoły developerskie pracujące nad aplikacjami PHP (np. w Laravel/Symfony), które chcą wdrożyć standaryzację środowisk przy użyciu Dockera i Docker Compose.
Minimalna liczba uczestników	3
Maksymalna liczba uczestników	8
Data zakończenia rekrutacji	06-03-2026
Forma prowadzenia usługi	zdalna w czasie rzeczywistym
Liczba godzin usługi	13
Podstawa uzyskania wpisu do BUR	Certyfikat systemu zarządzania jakością wg. ISO 9001:2015 (PN-EN ISO 9001:2015) - w zakresie usług szkoleniowych

Cel

Cel edukacyjny

Usługa przygotowuje uczestników do samodzielnej pracy z Dockerem w kontekście aplikacji PHP – od tworzenia obrazów i środowisk wielokontenerowych, przez debugowanie i optymalizację, aż po wdrożenia produkcyjne. Uczestnicy nauczą się stosować dobre praktyki bezpieczeństwa, wykorzystania wolumenów i sieci oraz integracji z procesami CI/CD, co pozwoli im efektywnie rozwijać i utrzymywać aplikacje webowe.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Uczestnik wyjaśnia różnicę między kontenerem a maszyną wirtualną.	Wskazuje główne cechy odróżniające kontenery od maszyn wirtualnych	Test teoretyczny
Uczestnik omawia architekturę Dockera i cykl życia kontenera.	Rozpoznaje elementy architektury (daemon, CLI, registry) oraz etapy życia kontenera	Test teoretyczny
Charakteryzuje podstawowe polecenia Dockera i ich zastosowanie.	Dopasowuje polecenia (np. docker run, docker ps, docker exec) do ich funkcji.	Test teoretyczny
Wyjaśnia rolę Dockerfile i zasady optymalizacji obrazów.	Opisuje znaczenie Dockerfile i podać przykłady optymalizacji.	Test teoretyczny
Charakteryzuje możliwości Docker Compose i definiuje usługi w pliku YAML.	Rozpoznaje poprawne definicje usług w docker-compose.yml i wskazuje zależności między nimi.	Test teoretyczny
Opisuje proces wdrażania aplikacji PHP z wykorzystaniem Dockera i Compose.	Wskazuje główne kroki uruchomienia aplikacji (budowa obrazu, docker-compose up, konfiguracja wolumenów)	Test teoretyczny

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji zawiera opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument potwierdza, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji?

TAK

Pytanie 3. Czy dokument potwierdza zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

Program

Usługa prowadzona jest w godzinach dydaktycznych. Przerwy nie są wliczone w ogólny czas usługi rozwojowej. Harmonogram usługi może ulec nieznacznemu przesunięciu, ponieważ ilość przerw oraz długość ich trwania zostanie dostosowana indywidualnie do potrzeb uczestników szkolenia. Łączna długość przerw podczas szkolenia nie będzie dłuższa aniżeli zawarta w harmonogramie.

Szczegółowy harmonogram realizacji usługi zostanie dostosowany do potrzeb i możliwości uczestników a jego finalna wersja zostanie podana zgodnie z wymaganiami systemu BUR, przed rozpoczęciem realizacji usługi.

Zajęcia zostaną przeprowadzone przez ekspertów z wieloletnim doświadczeniem, którzy przekazują nie tylko wiedzę teoretyczną, ale także praktyczne wskazówki i najlepsze praktyki. Uczestnicy mają możliwość czerpania z jego wiedzy i doświadczeń.

1. Wprowadzenie do konteneryzacji

- Czym różni się kontener od maszyny wirtualnej.
- Dlaczego Docker stał się standardem.
- Podstawowe pojęcia: obraz, kontener, wolumen, sieć.

2. Podstawy Dockera

Teoria

- Architektura Dockera (daemon, CLI, registry).
- Cykl życia kontenera.
- Docker Hub i prywatne rejestry.

Ćwiczenia

- Instalacja i pierwsze polecenia (docker run, docker ps, docker logs, docker exec).
- Uruchomienie kontenera PHP i testowe środowisko developerskie.
- Praca z wolumenami i sieciami Dockera.

3. Tworzenie obrazów dla PHP (3h)

Teoria

- Budowa obrazu – rola **Dockerfile**.
- Optymalizacja wielowarstwowa.
- Oficjalne obrazy PHP i ich warianty (CLI, Apache, FPM).
- Włączanie rozszerzeń PHP.

Ćwiczenia

- Stworzenie własnego obrazu PHP z Composerem.
- Dodanie frameworka (np. Laravel/Symfony).
- Zbudowanie aplikacji PHP z Nginx + PHP-FPM.
- Porównanie "fat image" vs "lean image".

4. Docker Compose – środowisko aplikacji

Teoria

- Co daje Docker Compose.
- Definicja usług w pliku docker-compose.yml.
- Zależności między usługami (np. PHP ↔ MySQL ↔ Redis).

Ćwiczenia

- Skonfigurowanie środowiska wielokontenerowego:
 - app – PHP,
 - web – Nginx,
 - db – MySQL.
- Użycie .env do parametrów środowiskowych.
- Restart polityki, healthcheck, zależności depends_on.

5. Debugowanie i logowanie w kontenerach

Teoria + Praktyka

- Sprawdzanie logów (docker logs, docker-compose logs).
- Wchodzenie do kontenera (exec, attach).
- Mapowanie logów PHP/Nginx na hosta.

6. Zaawansowane tematy

Teoria

- Najlepsze praktyki bezpieczeństwa w Dockerze (rootless, secrets).
- Multi-stage builds dla PHP (budowanie z Composerem, testy, obraz produkcyjny).
- Cache w Dockerze.
- CI/CD z Dockerem – jak pipeline używa Dockera (krótki wstęp).

Ćwiczenia

- Multi-stage build dla aplikacji Laravel (development → production).
- Push/pull do prywatnego rejestru (DockerHub/GitHub Registry).
- Dodanie do pipeline'u GitHub Actions kroku budującego obraz.

7. Deployment w praktyce

- Uruchomienie aplikacji na VPS z Docker Compose.
- Konfiguracja wolumenów (np. dla MySQL).
- Aktualizacja aplikacji → podmiana obrazu → rollback.

8. Podsumowanie i Q&A

- Najczęstsze błędy przy pracy z Dockerem.
- Jak dalej rozwijać wiedzę (Kubernetes, orkiestracja).
- Dyskusja o problemach z realnych projektów uczestników.

Walidacja

Harmonogram

Liczba przedmiotów/zajęć: 0

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
Brak wyników.					

Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	1 560,00 PLN
Koszt przypadający na 1 uczestnika netto	1 560,00 PLN
Koszt osobogodziny brutto	120,00 PLN
Koszt osobogodziny netto	120,00 PLN

Prowadzący

Liczba prowadzących: 0

Brak wyników.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

PDF/skrypt 1–2 strony

Przykładowe pliki (ZIP lub repo)

Warunki uczestnictwa

Przed zapisaniem się na usługę, w celu potwierdzenia dostępności miejsca w grupie szkoleniowej, prosimy o kontakt pod numerem telefonu

34 387 16 73 lub 530 642 270

Podstawa zwolnienia z VAT:

1) art. 43 ust. 1 pkt 29 lit. c Ustawy z dnia 11 marca 2024 o podatku od towarów i usług - w przypadku dofinansowania w wysokości 100%

2) § 3 ust. 1 pkt. 14 Rozporządzenia Ministra Finansów z dnia 20 grudnia 2013 r. w sprawie zwolnień od podatku od towarów i usług oraz warunków stosowania tych zwolnień - w przypadku dofinansowania w co najmniej 70%

3) W przypadku braku uzyskania dofinansowania lub uzyskania dofinansowania poniżej 70%, do ceny usługi należy doliczyć 23% VAT

Warunki techniczne

1. Sprzęt:

- Komputer/laptop z minimum 8 GB RAM (zalecane 16 GB).
- Procesor 4-rdzeniowy lub szybszy.
- Wolne miejsce na dysku: min. 20 GB (zalecane 50 GB, dla obrazów i kontenerów).

2. Oprogramowanie:

- System operacyjny: Windows 10/11, macOS 11+, Linux (Ubuntu/Debian/Fedora).
- Zainstalowany Docker Desktop (Windows/macOS) lub Docker Engine (Linux).
- Zainstalowany Docker Compose (jeśli nie wbudowany w Docker Desktop).
- Przeglądarka do połączenia z platformą zdalną (Chrome, Edge, Firefox).

3. Połączenie internetowe:

- Stabilne, min. 10 Mbps, zalecane 50 Mbps dla komfortowej pracy z transmisją wideo i pobieraniem obrazów Docker.

4. Dodatkowe wymagania:

- Konto Docker Hub (opcjonalne, do pobierania obrazów).
- Edytor tekstu / IDE dla PHP (np. VS Code).

Kontakt



Agata Flak

E-mail kontakt@dofinansowanekursy.pl

Telefon (+48) 791 511 221