

Technologie i oprogramowanie chmurowe

Projekt

Bazan Michał – 163881

Temat:

Konteneryzacja – Docker. Różnica między Docker Compose i Docker Swarm (konfigurowania wielu kontenerów na tym samym hoście, narzędzie do orkiestracji kontenerów). Docker + WSL2 - windows subsystem for linux.

Charakterystyka+porównanie+przykład implementacji.

Spis treści

Konteneryzacja.....	3
Czym jest konteneryzacja?.....	3
Konteneryzacja i Docker.....	3
Docker Compose i Docker Swarm.....	4
Docker Compose.....	4
Docker Swarm.....	4
Porównanie narzędzi.....	4
Implementacja Docker Compose.....	5
Implementacja Docker Swarm.....	5
Wnioski.....	5
Integracja z WSL2.....	6
Czym jest WSL2?.....	6
Test działania Dockera wewnątrz WSL2.....	6
Zalety i wady rozwiązania.....	6

Konteneryzacja

Czym jest konteneryzacja?

Konteneryzacja [1] to metoda wytwarzania, wdrażania i uruchamiania aplikacji w izolowanych środowiskach zwanych kontenerami. Kontenery są swoistą formą wirtualizacji, która umożliwia pakowanie aplikacji wraz z jej zależnościami i środowiskiem uruchomieniowym, co pozwala na jednolite i niezawodne działanie aplikacji na różnych platformach.

Izolacja, która charakteryzuje kontenery, sprawia, że aplikacja w kontenerze ma ograniczony dostęp do zasobów systemowych hosta, takich jak pamięć RAM, procesor czy dysk twardy. Dzięki temu kontenery mogą współdzielić ten sam system operacyjny hosta, ale działają w izolowanych środowiskach, co zapewnia lepszą wydajność i bezpieczeństwo w porównaniu z tradycyjnymi maszynami wirtualnymi.

Kontenery są zwykle oparte na obrazach kontenerowych, które zawierają wszystko, co potrzebne do uruchomienia aplikacji, takie jak kod aplikacji, środowisko uruchomieniowe, biblioteki i inne zależności. Dzięki temu kontenery są przenośne i można je łatwo wdrażać w różnych środowiskach, co pozwala deweloperom na konsekwentne i niezawodne dostarczanie aplikacji.

Konteneryzacja staje się coraz bardziej popularna w dziedzinie rozwoju oprogramowania ze względu na swoją elastyczność, niezawodność i możliwość automatyzacji procesów wdrażania i skalowania aplikacji.

Konteneryzacja i Docker

Docker [2] [3] to potężne narzędzie, które rewolucjonizuje sposób, w jaki deweloperzy, inżynierowie DevOps i administratorzy zarządzają aplikacjami i infrastrukturą IT. Dzięki Dockerowi, aplikacje są pakowane w kontenery, które zawierają wszystkie potrzebne zależności, biblioteki i środowisko uruchomieniowe, co sprawia, że są przenośne i działają tak samo w każdym środowisku, niezależnie od tego, czy jest to środowisko deweloperskie, testowe czy produkcyjne.

Jego elastyczność pozwala na szybkie tworzenie, uruchamianie, skalowanie i zarządzanie aplikacjami w kontenerach. Deweloperzy mogą łatwo tworzyć lokalne środowiska deweloperskie, wdrażać aplikacje na różnych platformach chmurowych i zarządzać skalowaniem aplikacji w zależności od zmieniających się potrzeb biznesowych.

Jednym z kluczowych atutów Docker jest także możliwość integracji z innymi narzędziami i technologiami, takimi jak Continuous Integration/Continuous Deployment (CI/CD) pipeline'y, narzędzia monitorowania i logowania, czy też narzędzia do zarządzania infrastrukturą jako kodem.

Dodatkowo, Docker cieszy się dużą popularnością dzięki swojej otwartej społeczności, która stale rozwija ekosystem narzędzi, obrazów i rozwiązań, co ułatwia pracę z Dockerem i zapewnia wsparcie dla szerokiego zakresu zastosowań, od prostych aplikacji internetowych po złożone systemy mikroserwisów.

Docker Compose i Docker Swarm

Docker Compose

Docker Compose [4] jest narzędziem służącym do definiowania i uruchamiania wielokontenerowych aplikacji. Pozwala ono deweloperom opisać konfigurację wielu kontenerów w plikach YAML, co ułatwia zarządzanie zależnościami między kontenerami oraz uruchamianie aplikacji w środowiskach deweloperskich i testowych. Docker Compose jest szczególnie użyteczny w prostych aplikacjach, które składają się z kilku kontenerów i nie wymagają zaawansowanego skalowania.

Docker Swarm

Docker Swarm [5] jest narzędziem do orkiestracji kontenerów, które umożliwia zarządzanie klastrami kontenerów w środowiskach produkcyjnych. Pozwala na tworzenie i zarządzanie wieloma kontenerami na wielu maszynach fizycznych lub wirtualnych, co zapewnia skalowalność i wysoką dostępność aplikacji. Docker Swarm oferuje funkcje takie jak równoważenie obciążenia, odtwarzanie po awarii, aktualizacje bez przestoju oraz monitorowanie stanu klastra. Jest to idealne narzędzie do budowania i zarządzania złożonymi systemami mikroservisów w środowiskach produkcyjnych.

Porównanie narzędzi

Docker Compose i Docker Swarm to dwa różne narzędzia stworzone przez Docker, Inc., które pomagają w zarządzaniu kontenerami, ale posiadają różne funkcje i zastosowania.

Zastosowanie: Docker Compose jest skoncentrowany na uruchamianiu wielokontenerowych aplikacji w środowiskach deweloperskich i testowych, podczas gdy Docker Swarm jest przeznaczony do zarządzania klastrami kontenerów w środowiskach produkcyjnych.

Skalowanie: Docker Swarm oferuje zaawansowane funkcje skalowania, takie jak automatyczne równoważenie obciążenia i skalowanie horyzontalne, które są nieobecne w Docker Compose.

Złożoność: Docker Compose jest prostszy w użyciu i szybszy do nauki, podczas gdy Docker Swarm może być bardziej złożony ze względu na potrzebę zarządzania klastrami i konfiguracją wielu węzłów.

Elastyczność: Docker Compose jest bardziej elastyczny i bardziej odpowiedni dla prostych aplikacji, podczas gdy Docker Swarm jest bardziej skomplikowany, ale oferuje większą kontrolę nad zasobami i skalowaniem aplikacji.

Implementacja Docker Compose

Implementacja Docker Swarm

Wnioski

Integracja z WSL2

Czym jest WSL2?

Test działania Dockera wewnątrz WSL2

Zalety i wady rozwiązania

Źródła

- [1]. <https://pl.wikipedia.org/wiki/Konteneryzacja>
- [2]. [https://pl.wikipedia.org/wiki/Docker_\(oprogramowanie\)](https://pl.wikipedia.org/wiki/Docker_(oprogramowanie))
- [3]. <https://docs.docker.com/>
- [4]. <https://docs.docker.com/compose/>
- [5]. <https://docs.docker.com/engine/swarm/>