



Docker

Co to jest Docker?

Programistyczna platforma kontenerowa, zaprojektowana z myślą o tworzeniu, dostarczaniu i uruchamianiu aplikacji z użyciem technologii kontenerowej. Docker jest dostępny w dwóch wersjach: biznesowej i społecznościowej

Jak działa Docker?

Docker działa na zasadzie tworzenia kontenerów - izolowanych środowisk uruchomieniowych dla aplikacji i jej zależności. Każdy kontener korzysta z jednego lub kilku obrazów, które zawierają wszystkie niezbędne zależności oraz konfiguracje do uruchomienia aplikacji. Kontenery są izolowane od siebie oraz od hosta, co pozwala na uniknięcie problemów związanych z różnicami w konfiguracji systemów operacyjnych.

Historia Docker

Początkowo Docker był wewnętrznym projektem rozwijanym w firmie dotCloud przez Solomona Hykesa i Sebastiena Pahla. Pierwszy raz został uruchomiony w 2011 roku, natomiast w marcu 2013 roku Docker został udostępniony publicznie podczas konferencji PyCon w Santa Clara. Od tego czasu zyskuje coraz większą popularność, jego profil na portalu GitHub został oznaczony gwiazdką prawie 50 tysięcy razy, a członkowie społeczności dokonali ponad 15 tysięcy forków (rozwidleń) projektów.

Zalety korzystania z Docker

Docker oferuje wiele korzyści w porównaniu do tradycyjnych metod wdrażania aplikacji. Najważniejsze z zalet to:

- Izolacja aplikacji i jej zależności
- Łatwość w tworzeniu i dystrybucji aplikacji
- Skalowalność aplikacji w zależności od potrzeb
- Elastyczność i łatwość w modyfikacji środowiska aplikacji
- Możliwość korzystania z gotowych obrazów Docker, co oszczędza czas i usprawnia proces wdrażania aplikacji

CaaS

Containers as a Service (CaaS) lub Container Services to zarządzane usługi chmurowe, zarządzające cyklem życia kontenerów. Usługi kontenerowe pomagają organizować (rozpoczynać, zatrzymywać i skalować) środowisko uruchomieniowe kontenerów. Korzystając z usług kontenerowych, można upraszczać, automatyzować i przyspieszać cykl tworzenia i wdrażania aplikacji.

Czym jest kontener?

Kontener Docker to format pakowania, gromadzący całość kodu i zależności aplikacji w standardowym formacie, pozwalającym na szybkie i niezawodne działanie we wszystkich środowiskach obliczeniowych. Kontener Docker jest popularnym, lekkim, autonomicznym i wykonywalnym kontenerem, zawierającym wszystko, co jest potrzebne do uruchomienia aplikacji, w tym biblioteki, narzędzia systemowe, kod i środowisko uruchomieniowe. Kontenery są nietrwałe i są uruchamiane za pomocą obrazów.

Różnica między kontenerem a maszyną wirtualną

W przeciwieństwie do maszyny wirtualnej (VM), która zapewnia wirtualizację sprzętową, kontener zapewnia lekką wirtualizację na poziomie systemu operacyjnego poprzez abstrahowanie „przestrzeni użytkownika”. Kontenery dzielą jądro systemu hosta z innymi kontenerami. Kontener, który działa na systemie operacyjnym hosta, jest standardową jednostką programową, gromadzącą kod i wszystkie jego zależności, dzięki czemu aplikacje mogą działać szybko i niezawodnie pomiędzy różnymi środowiskami.

Silnik Docker

Oprogramowanie open source hosta do tworzenia i uruchamiania kontenerów. Silniki Docker działają jako aplikacja klient-serwer, obsługująca kontenery na różnych serwerach Windows i systemach operacyjnych Linux, w tym Oracle Linux, CentOS, Debian, Fedora, RHEL, SUSE i Ubuntu.

Obrazy Docker

Zbiór oprogramowania uruchamianego jako kontener, zawierający zbiór instrukcji tworzenia kontenera, który może działać na platformie Docker. Obrazy są niezmiennie, a zmiany obrazu wymagają utworzenia nowego obrazu. Tworzenie obrazu Docker ze wszystkimi zależnościami rozwiązuje problem „ale przecież to działało na mojej maszynie do programowania”. Kluczową kwestią jest automatyczne tworzenie obrazu Docker w potoku kompilacji z repozytorium kodu źródłowego (np. Git) i początkowe testowanie go w środowisku programistycznym. Ten niezmienny obraz zostaje potem zapisany w rejestrze Docker.

Rejestr Docker

Obrazy Docker są łatwe w budowie i programiści uwielbiają ich prostotę oraz przenośność, szybko odkryto, że zarządzanie tysiącami obrazów Docker jest bardzo trudnym zadaniem. Rejestr Docker jest odpowiedzią na to wyzwanie. Rejestr Docker to miejsce przechowywania i pobierania obrazów. Jest bezstanową i skalowalną aplikacją po stronie serwera, składującą i dystrybuującą obrazy Docker.

Komponenty Docker

Podstawowe i najczęściej wykorzystywane komponenty Docker to:

- Docker Engine
- Docker Hub
- Docker Compose
- Docker Swarm
- Docker Machine
- Docker CLI
- Docker file

Komponenty Docker – krótki opis

- Docker Engine – jest to główny komponent Dockera, który zarządza kontenerami i ich wdrożeniem. To podstawowe narzędzie, które umożliwia tworzenie, uruchamianie i zarządzanie kontenerami.
- Docker Hub – jest to publiczny rejestr kontenerów Dockera, w którym można przechowywać, udostępniać i wyszukiwać kontenery Dockera. Dzięki Docker Hub można łatwo dzielić się kontenerami z innymi użytkownikami Dockera.
- Docker Compose – jest o narzędzie, które umożliwia łatwe uruchamianie wielu kontenerów Dockera w jednym środowisku. Dzięki temu można łatwo skonfigurować i uruchomić złożone aplikacje, które wymagają kilku kontenerów Dockera.

Komponenty Docker – krótki opis

- Docker Swarm – jest to narzędzie do zarządzania klastrami kontenerów Dockera, które umożliwia skalowanie i zarządzanie aplikacjami, które działają w wielu kontenerach Dockera. Dzięki Docker Swarm można łatwo zarządzać wieloma kontenerami Dockera jako jednym systemem.
- Docker Machine – jest to narzędzie do łatwego tworzenia i zarządzania wirtualnymi maszynami Dockera. Dzięki Docker Machine można łatwo uruchamiać i zarządzać kontenerami Dockera na różnych systemach operacyjnych i w różnych środowiskach.

Komponenty Docker – krótki opis

- Docker CLI – jest to interfejs wiersza poleceń Dockera, który umożliwia programistom i deweloperom łatwe zarządzanie kontenerami Dockera za pomocą poleceń w konsoli.
- Dockerfile – jest to plik konfiguracyjny Dockera, który umożliwia definiowanie konfiguracji i środowiska kontenera. Dzięki Dockerfile można łatwo tworzyć i zarządzać kontenerami Dockera za pomocą kodu źródłowego.

Komponenty Docker

Wymienione komponenty to najbardziej podstawowe narzędzia do budowy, wdrażania i zarządzania aplikacjami w kontenerach Docker. Oprócz tego, Docker oferuje wiele innych narzędzi i komponentów, które są wykorzystywane w bardziej zaawansowanych przypadkach użycia.

Komponenty Docker

Dodatkowe przykładowe narzędzia używane w bardziej zaawansowanych przypadkach użycia Docker:

- Docker Registry
- Docker Enterprise
- Docker Toolbox
- Docker Network
- Docker Volume
- Docker API
- i wiele więcej

Przykłady użycia Docker

Docker może być używany w wielu przypadkach, w tym między innymi:

- Wdrażanie aplikacji webowych, takich jak serwery WWW czy aplikacje e-commerce
- Uruchamianie baz danych, takich jak MySQL czy PostgreSQL
- Testowanie aplikacji w izolowanych środowiskach
- Konteneryzacja mikroservisów w architekturze mikrousług

Docker, a rozwój oprogramowania

Kontenery mogą być pakowane przez dowolnego programistę. Osoby z branży oprogramowania często grupują programistów według ich specjalizacji – frontend, backend lub dowolnej pośredniej kombinacji. Pomimo tego, że w większości przypadków pakowaniem kontenerów zajmują się programiści backendu, każda osoba zaznajomiona z podstawowymi pojęciami CaaS, może odnieść sukces w tym konkretnym obszarze cyklu rozwoju oprogramowania.

Docker jako środowisko uruchomieniowe mikrouslug

Koncept pocięcia monolitycznych aplikacji na mniejsze kawałki mikrouslug spotyka się w dzisiejszych czasach z dużym zainteresowaniem wśród programistów. Mikrouslugi są wdrażane niezależnie jako procesy, używają lekkich protokołów do komunikowania się między sobą, a każda usługa posiada własne dane. Ponieważ mikrouslugi stosują zdecentralizowane podejście do zarządzania, wymagają one raczej wysokiego stopnia automatyzacji infrastruktury, automatycznego testowania, w pełni zautomatyzowanych potoków ciągłego dostarczania (CD) oraz wykwalifikowanych, zwinnych zespołów DevOps.

Kubernetes

Platforma chmurowa Kubernetes jest oprogramowaniem stworzonym za pomocą Docker. Jest to otwarty system do automatyzacji wdrażania, skalowania i zarządzania aplikacjami kontenerowymi. Został stworzony przez Google i obecnie jest rozwijany przez fundację CNCF (Cloud Native Computing Foundation). Umożliwia zarządzanie tysiącami kontenerów w skali globalnej i jest używany przez wiele firm, w tym takie giganty jak Airbnb, Spotify czy Uber. Kubernetes wykorzystuje kontenery Docker jako jednostkę podstawową do uruchamiania aplikacji oraz pozwala na automatyzację całego procesu wdrażania i zarządzania aplikacjami.

Apache Spark

Platforma do analizy danych Apache Spark jest oprogramowaniem stworzonym przy użyciu Docker. Jest to otwarty system przetwarzania dużych zbiorów danych, który umożliwia wykonywanie złożonych operacji na danych w czasie rzeczywistym. Apache Spark wykorzystuje kontenery Docker do izolowania środowiska uruchomieniowego i wersji bibliotek, co pozwala na łatwe przenoszenie i uruchamianie aplikacji między różnymi środowiskami. Wykorzystuje też wiele narzędzi dostępnych w ekosystemie Docker, takich jak Docker Compose czy Docker Swarm, co ułatwia wdrażanie aplikacji w kontenerach. Umożliwia wykonywanie złożonych operacji na dużych zbiorach danych w czasie rzeczywistym, co pozwala na szybkie podejmowanie decyzji i reagowanie na zmiany w danych. Apache Spark jest używany przez wiele firm, w tym takie giganty jak Netflix, Uber czy Yahoo.

Uruchamiane w Docker

Docker może posłużyć do uruchamiania podstawowych narzędzi tworzenia oprogramowania oraz aplikacji za pomocą kontenerów. Takie uruchomienie pozwala na szybkie rozpoczęcie pracy bez długiego czasu ustawiania programów pod konkretny projekt. Przykładami takich narzędzi są:

- WordPress
- Jenkins
- GitLab
- I wiele innych

Wersje Docker i jego wydania

Tempo wydań Docker jest o wiele szybsze niż cykl wydawniczy tradycyjnego oprogramowania dla firm. Czasami sama szybkość wydań Docker wraz ze świeżością projektu Docker wzbudza obawy dotyczące jego bezpieczeństwa i stabilności. Mimo że Docker i jego linia poleceń, proces daemon Docker, jego API i narzędzia, takie jak Docker Swarm, Docker Machine i Docker Compose powstały jedynie w ciągu ostatnich trzech lat, podstawowe funkcje jądra są dostępne w każdym jądrze systemu Linux od niemal dekady. Wybitnym przykładem wczesnego użytkownika technologii kontenerowej jest Google. Google używało kontenerów Linux zanim pojawił się Docker. Co więcej, w Google wszystko działa w kontenerze. Szacuje się, że Google uruchamia kilka miliardów kontenerów tygodniowo.

Film podsumowujący o podstawach Docker

Film w języku angielskim na przykładzie aplikacji w Node.js – 11 minut

https://www.youtube.com/watch?v=gAkwW2tulqE&ab_channel=Fireship