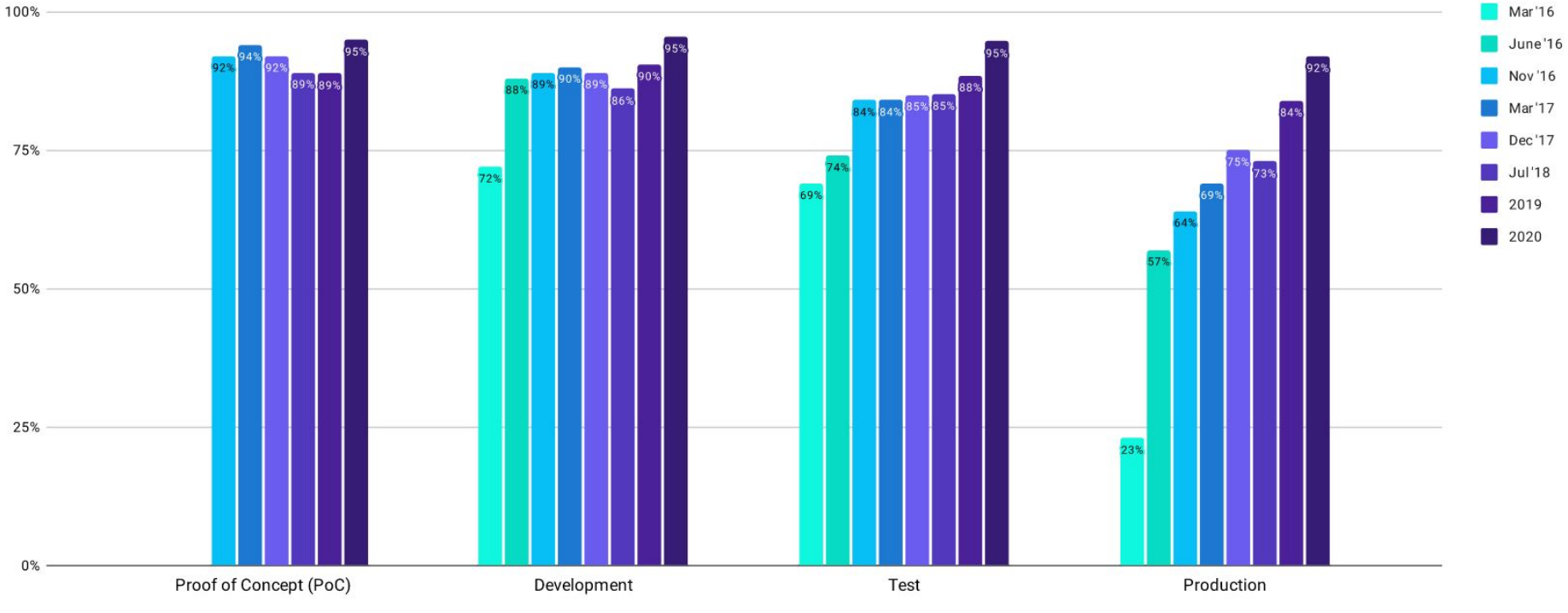
Konteneryzacja

Agenda

- Wstęp
- Wirtualizacja i konteneryzacja
- Obrazy kontenerów
- Docker - charakterystyka
- zarządzanie kontenerami
- zadania

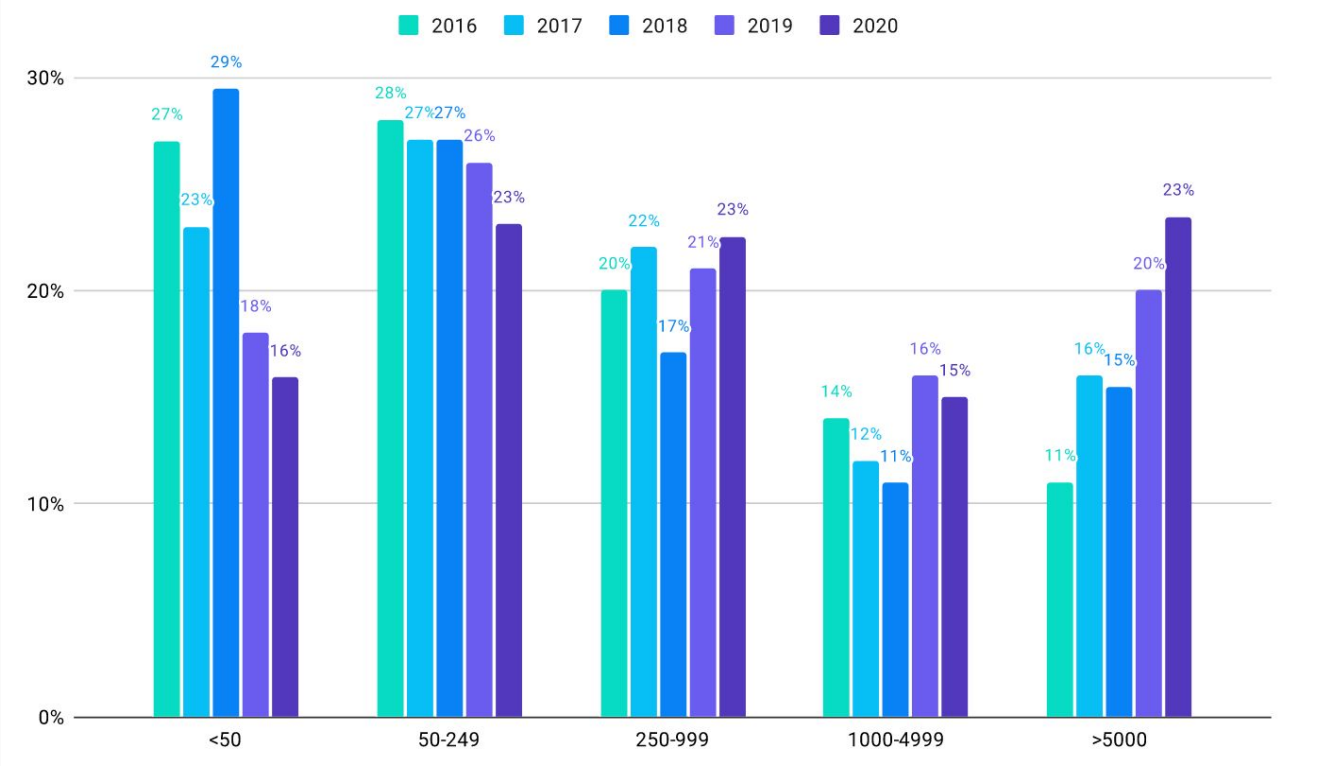
Wstęp - statystyka CNCF 2020

Please indicate if your company/organization currently uses, or has future plans to use, containers for any of the below options



Wstęp - statystyka CNCF 2020

How many containers does your company/organization typically run?

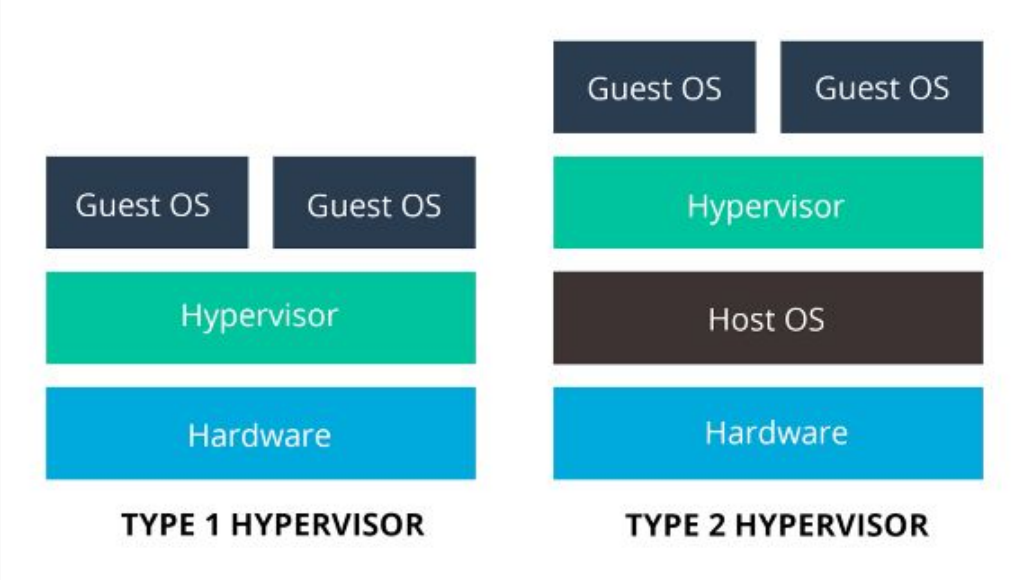


Wirtualizacja

- metoda uruchamiania wirtualnego środowiska bezpośrednio na sprzęcie fizycznym innego hosta - storage, RAM, CPU itp.
- jednostka wirtualizacji - maszyna wirtualna
- na jednym serwerze może zostać uruchomionych wiele maszyn wirtualnych
- maszyny wirtualne mają odseparowane zasoby
- do zarządzanie hardwarem i przydzielania zasobów wykorzystywany jest specjalny komponent - **Hypervisor** (np. VirtualBox, KVM, Proxmox)

Hypervisor - rodzaje

- Type 1 / bare metal
- Type 2 / hosted



Maszyny wirtualne - use cases

- aplikacje wrażliwe na duże obciążenie
- środowiska dev/test
- centra danych on-premise
- środowiska chmurowe

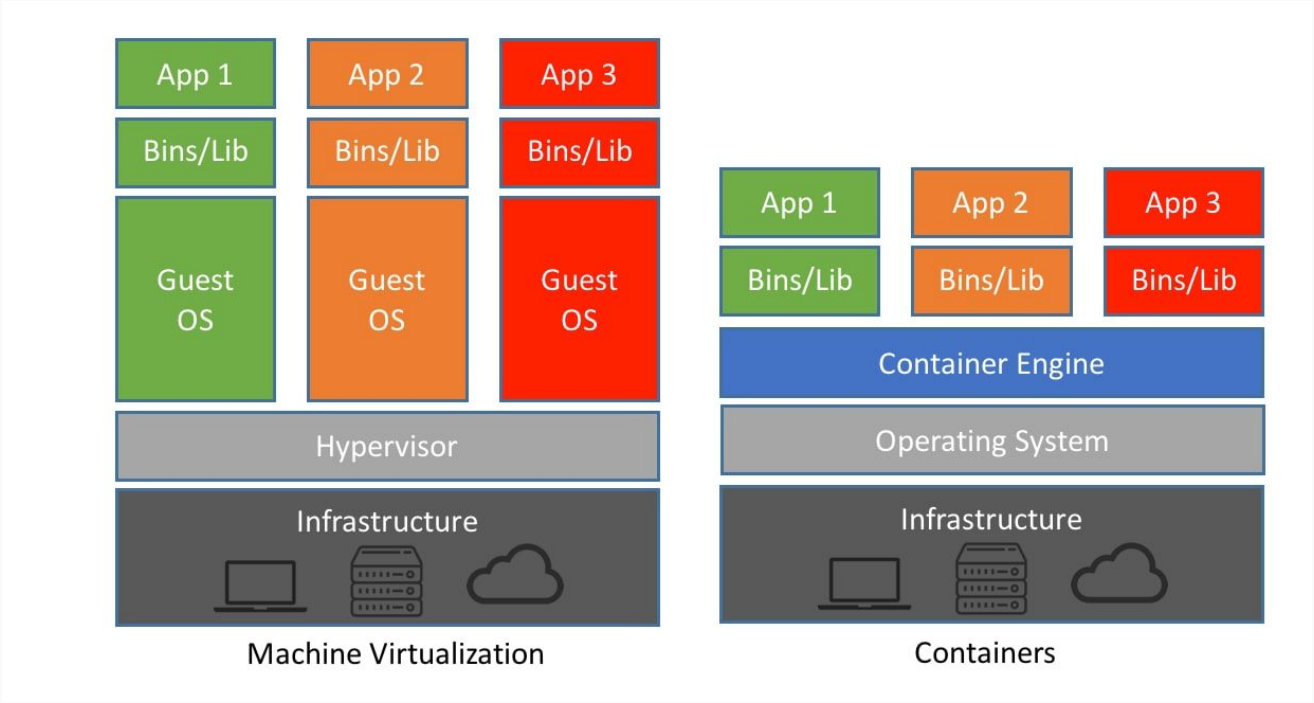
Maszyny wirtualne - problemy

- duże zużycie zasobów
- długi czas startu
- konieczność wykorzystywania hypervisora
- niska utylizacja zasobów dla małych aplikacji

Konteneryzacja

- Metoda wirtualizacji
- mniejsze zużycie zasobów w porównaniu do maszyn wirtualnych
- Kontenery korzystają z tego samego kernela systemu operacyjnego
- Każdy kontener ma własny system plików i zmienne środowiskowe
- Container Engine (Container Runtime) - aplikacja zarządzająca kontenerami na serwerze, np. docker, containerd, cri-o

Wirtualizacja vs konteneryzacja



Wirtualizacja vs konteneryzacja

VMS	CONTAINERS
Heavyweight	Lightweight
Each VM runs in its own OS	All containers share the host OS
Hardware-level virtualization	OS virtualization
Startup time in minutes	Startup time in milliseconds
Allocates required memory	Requires less memory space
Fully isolated and hence more secure	Process-level isolation, possibly less secure

Dlaczego warto używać konteneryzacji?

- jedno narzędzie do uruchamiania różnych aplikacji
- zwiększa szybkość wytwarzania oprogramowania
- niezależne od technologii aplikacji
- wykorzystywane wszędzie

Kontenery - cechy

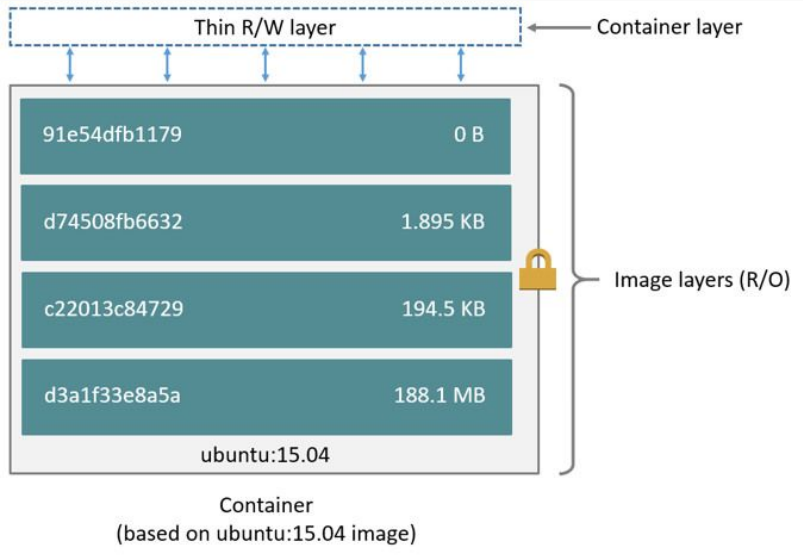
- ustandaryzowane
- przenośne
- niezależne od innych kontenerów
- skalowalne
- lekkie
- bezstanowe

Konteneryzacja - słownik pojęć

- **obraz** - plik binarny utworzony według określonego standardu kontenerów (Open Container Initiative - OCI)
- **kontener** - uruchomiona instancja obrazu
- **host** - system na którym uruchomiony jest Container Engine

Struktura obrazów

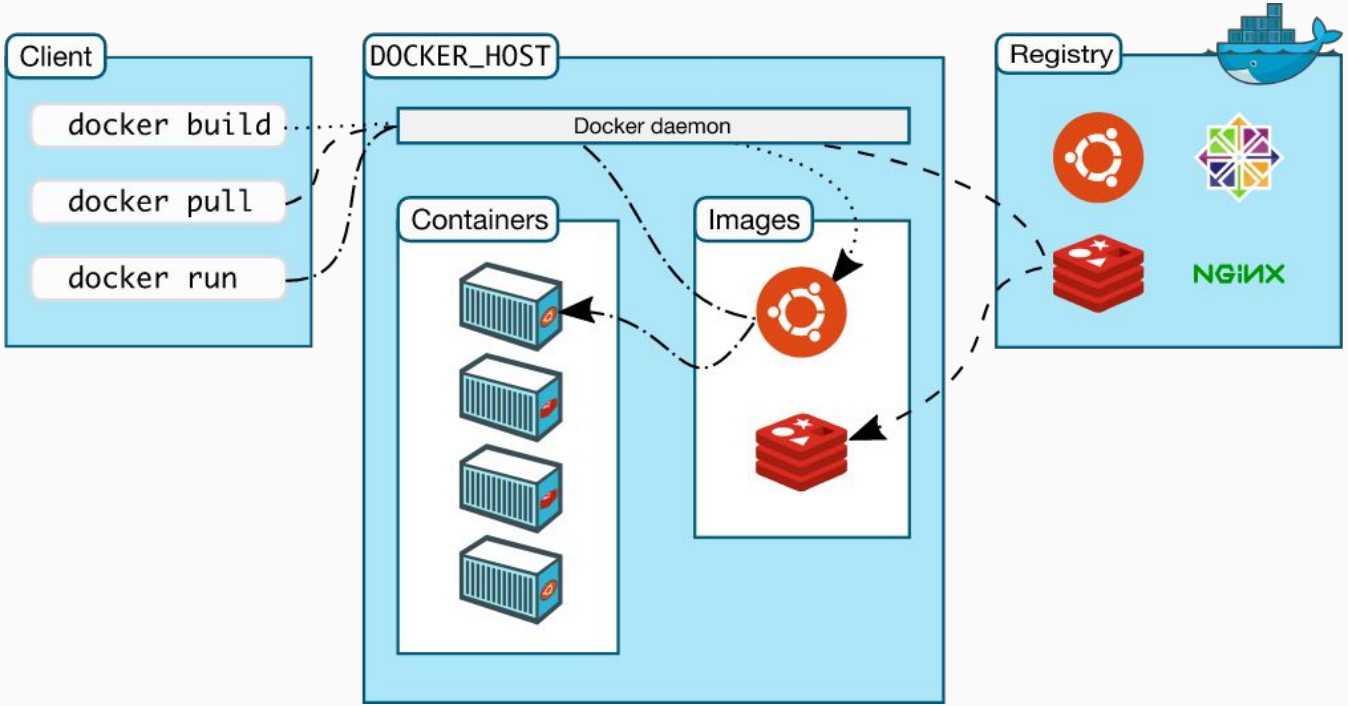
- podczas budowania obrazów tworzone obrazy pośrednie (struktura warstwowa)
- kontener jest obrazem składającym się z niezależnej warstwy read/write oraz uporządkowanego zbioru warstw read-only na niższych poziomach
- warstwy read-only mogą być współdzielone przez różne kontenery, ale tylko dla operacji do odczytu
- w wypadku konieczności edycji pliku z warstw read-only tworzona jest kopia w warstwie kontenera i to na niej przeprowadzane są operacje (Copy on write strategy)
- kiedy potrzebne są trwałe, współdzielone zasoby wykorzystuje się docker volume



Docker

- Najpopularniejsza platforma do tworzenia i uruchamiania kontenerów
- Docker Engine - aplikacja składająca się z serwera (Docker daemon) oraz klienta (Docker CLI - komenda docker) komunikujących się przez REST API
- serwery Docker daemon mogą współpracować ze sobą w celu tworzenia serwisów - Docker Swarm

Architektura Dockera



Obrazy Dockera

- Obrazy Dockera tworzone są w oparciu o plik **Dockerfile**
- Dockerfile składa się ze zbioru instrukcji wykonywanych kolejno podczas tworzenia obrazu
- do stworzenia obrazu wykorzystywana jest komenda **docker build**
- utworzone obrazy są przechowywane w lokalnym systemie plików. wysłanie ich do zewnętrznego repozytorium jest wykonywane za pomocą komendy **docker push**.
- uruchamianie kontenerów: **docker run**
- publiczne repozytorium obrazów: Docker Hub, prywatne: Nexus

Docker Compose

- Narzędzie do tworzenia aplikacji składających się z wielu kontenerów
- opiera się o plik YAML w którym określane jest jakie kontenery mają zostać uruchomione
- do włączenia całej aplikacji wystarczy wywołać jedną komendę: docker-compose up
- wszystkie uruchomione kontenery są domyślnie w jednej sieci
- docker compose działa lokalnie tzn. nie synchronizuje informacji pomiędzy różnymi hostami

Docker Compose - cechy

- Pozwala utworzyć wiele izolowanych środowisk dockerowych na jednym hoście
- umożliwia deklaratywne zarządzanie środowiskiem - plik **docker-compose.yml**
- zachowuje dane wolumenów podczas tworzenia kontenerów
- podczas uruchamiania restartuje tylko te kontenery które się zmieniły
- wspiera zarządzanie zmiennymi środowiskowymi pomiędzy kontenerami
- idealny do lokalnego developmentu
- rzadko wykorzystywany jako środowisku produkcyjne

Źródła

- https://www.cncf.io/wp-content/uploads/2020/12/CNCF_Survey_Report_2020.pdf
- https://www.redhat.com/en/topics/virtualization/what-is-a-virtual-machine
- https://medium.com/teamresellerclub/type-1-and-type-2-hypervisors-what-makes-them-different-6a1755d6ae2c
- https://www.redhat.com/en/resources/virtualization-use-cases-technology-overview
- https://www.netapp.com/blog/containers-vs-vms/
- https://www.backblaze.com/blog/vm-vs-containers/
- https://docs.docker.com/storage/storagedriver/
- https://docs.docker.com/get-started/overview/

Zad 1 - instalacja środowiska

Należy uruchomić środowisko do lokalnego developmentu. Wymagane aplikacje:

- relacyjna baza danych - PostgreSQL 13.5
- baz NoSQL - MongoDB
- broker wiadomości - RabbitMQ

Zad 2 - aplikacja

Należy stworzyć aplikację Spring Boot i połączyć z usługami uruchomionymi w poprzednim zadaniu.

Przetestować połączenie:

- PostgreSQL - zapis i odczyt
- MongoDB - zapis i odczyt
- RabbitMQ - wysyłanie i odbieranie wiadomości

Zad 3 - konteneryzacja aplikacji

Należy utworzyć obraz uruchamiający aplikację Spring Boot utworzoną w poprzednim zadaniu, a następnie ją uruchomić w utworzonym środowisku.

Tworzenie obrazu na dwa sposoby:

- manualnie przez Dockerfile
- poprzez wbudowany mechanizm Spring Boot