

1. Witam wszystkich
 1. w tej prezentacji chciałbym powiedzieć trochę o Dockerze, trochę o wirtualizacji i konteneryzacji danych.
2. Co to jest konteneryzacja i dlaczego jest lepsza od wirtualizacji
 1. zacznijmy od tego co to jest wirtualizacja:
 1. większość z was wie co to znaczy postawić wirtualną maszynę, albo coś w tym stylu. To dokładnie jest wirtualizacja, czyli:
 1. Wykorzystując fizyczne miejsce na dysku tworzymy coś co nie do końca jest fizyczne, tzn. możemy na przykładowo Windowsie stworzyć wirtualny system np. Linux.
 2. wiemy jednak, że nawet bardzo dobry komputer przy wirtualizacji będzie się trochę wieszał i nie działał tak jakby taki system byłby postawiony bez wirtualizacji
 2. Obrazek na dole pokazuje nam jak działa wirtualizacja. kilka aplikacji w jednym miejscu, na jednym systemie.
 2. Teraz popatrzymy na obrazek i porównajmy. Jak widać wirtualna maszyna posiada:
 1. server
 2. Jakiś sytem pierwotny
 3. hypervisor. Supervisor to nadzorca naszego systemu, za to hypervisor to nadzorca pracy systemu wirtualnego. hypervisor działa tak, że sprawdza które systemy z naszej fizycznej maszyny mogą być użyte przez maszyny wirtualne i je używa.
 4. następnie widzimy jakieś systemy wirtualne, nad nimi biblioteki i w końcu jakaś baza danych lub aplikacja z której chcemy korzystać.
 3. Obok widać schemat działania konteneryzacji. Co to jest? Alternatywa dla wirtualizacji, tzn. zamiast hypervisor mamy silnik kontenerowy. Jest on z góry przypisany na system Linux i tutaj już oszczędzamy jeden krok. Nie potrzebujemy wirtualizować całego systemu operacyjnego, przez takie uproszczenie każdy kontener działa dokładnie na takiej ilości pamięci jaką potrzebuje, nie zabiera nam RAM'u na system, który jest tylko po to żeby aplikacja uruchomić.
 1. powyżej widzimy poszczególne kontenery, kilka kontenerów z aplikacją, kilka z bazą danych. Dzięki temu, że możemy łączyć konkretne kontenery nie musimy instalować/ wrzucać do każdego biblioteki. Następna rzecz, która ułatwia nam życie, a maszynie oszczędza pamięć.
3. Czym jest Docker?
 1. Docker to w tym momencie najpopularniejszy system do konteneryzacji danych.
 2. Kontenery Dockera zawierają wszystko czego potrzebuje aplikacja do działania, a aplikacja działa w tym kontenerze także nie musimy się przejmować czy zadziała nam na innym komputerze. Działa i tyle.
 3. Mała waga:
 1. kontenery działające na tej samej maszynie i systemie, w tym przypadku KERNEL uruchamiają się w mgnieniu oka i zabierają mniej RAM'u. Obrazy są konstruowane na warstwowych systemach plików i działają na tych samych plikach, powodując, że użycie dysku oraz pobieranie obrazów jest bardziej efektywne.
 4. Otwieranie:
 1. kontenery dockera są zrobione w standardach linuxowych oraz bez problemu otwierają się na windowsie.
 5. Bezpieczeństwo:
 1. kontenery izolują aplikację od innych ale również od dolnych struktur, przez co nic nie powinno się stać ani z aplikacją, ani z innymi aplikacjami nawet na tym samym systemie.

6. Tworzenie lepszych systemów:

1. Docker zapewnia że nie będziemy musieli, jako programiści, skupiać się na tworzeniu odpowiedniego środowiska programistycznego, a zamiast tego będziemy mogli skupić się na tym co najważniejsze, czyli Tworzeniu aplikacji, poprawianiu usterek oraz udostępnianiu tego co zrobimy.
2. "Skończmy godzinami ustawiać środowisko programistyczne..." Docker pokazuje nam jak proste jest używanie aplikacji, które sobie umieścimy w nim. Wrzucamy aplikację do kontenera i otwieramy ją wszędzie gdzie tylko chcemy na maszynie na której dział docker.
3. "Zwiększ kreatywność". Dzięki temu że docker istnieje możemy bez żadnych problemów zwiększyć procent pracy nad aplikacjami, zamiast martwić się nad instalowaniem koniecznych systemów na naszych komputerach aby je po prostu uruchomić. Przykładowo octave, Aby zaistalować octave na macu musiałem zaistalować jeszcze dodatkowo virtualbox, xquartz i w sumie nie bardzo wiem po co, ale bez tego nie pójdzie. Tutaj tego unikniemy.
4. to samo dotyczy konfigurowania naszego oprogramowania programistycznego, tak aby działało tak jak u twórcy. Aplikacja, oraz wszystkie potrzebne biblioteki są już w danym kontenerze, a my po prostu to włączamy.

7. "share and collaborate":

1. Tak jak w tytule, dziel i współpracuj. Docker pozwala na działanie równolegle z swoimi współpracownikami.
2. Równie łatwe jest dzielenie się aplikacjami. Chcesz wpisać sobie do cv, że napisałeś super program na Projekcie z c++, użyj do tego docker'a. Bardzo dużo star upów korzysta z takiego narzędzia. nie musisz już martwić się tym jak pokażesz aplikację na rozmowie, skoro pisałeś ją na innym komputerze, albo nagle wywali Ci błąd. Wrzuć ją na docker'a i zawsze będzie działała tak samo.