Politechnika Śląska

Sprawozdanie z modułu nr 1

BRS 2022/2023

Bezpieczeństwo rozwiązań skonteneryzowanych

Kierunek: Informatyka

Członkowie zespołu:
Kamil Donda
Robert Kwoll

Gliwice, 2022/2023

Spis treści

1 Wprowadzenie

Konteneryzacja w świecie IT czerpie pełnymi garściami z idei kontenerów stosowanych w branży logistycznej. To nic innego, jak wirtualny zasobnik mieszczący najczęściej aplikację lub jej część, wraz z zestawem plików niezbędnych do uruchomienia kodu.

Co warte podkreślenia, aplikacja w takim zasobniku nie jest już tak mocno zależna od rodzaju i konfiguracji infrastruktury. Może być w prosty sposób przenoszona, powielana i wdrażana w różnych środowiskach (on premise, w chmurze publicznej, chmurze hybrydowej, etc.).

W branży IT kontenery umożliwiają odejście od tradycyjnej monolitycznej struktury tworzenia aplikacji. Dzielą je na mniejsze i łatwiejsze do zarządzania elementy – tzw. mikrousługi (ang. microservices). Dzięki temu każda z nich (np. funkcja w konkretnej aplikacji) może być łatwiej zaktualizowana, bezpieczniej przetestowana i szybciej wdrożona przez zespół DevOps. To szczególnie ważne kiedy standardem jest zapewnienie stałej dostępności do oferowanych usług i rozwiązań.

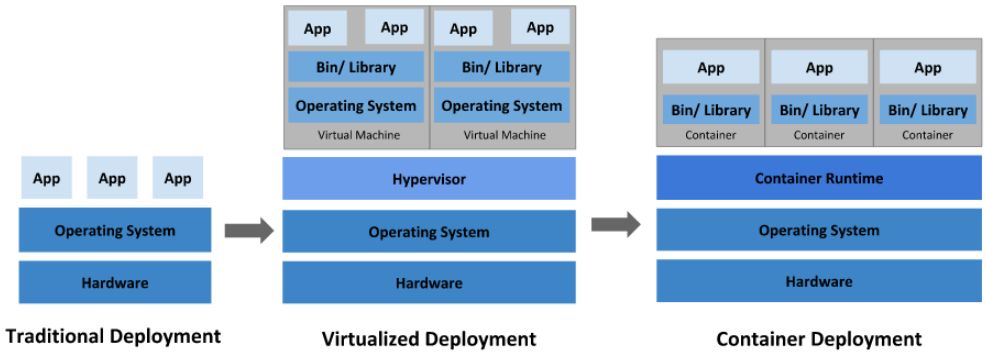
W środku takiego kontenera znajduje się zatem właściwy kod aplikacji lub jej części, a do tego wymagane biblioteki, procesy, zależności, pliki binarne oraz sam silnik służący do uruchomienia zawartości zasobnika.

Kontenery są coraz chętniej stosowane przy tworzeniu nowoczesnych środowisk aplikacyjnych. Należy jednak pamiętać, że nie stanowią one panaceum na wszystkie możliwe problemy. Bezpieczeństwo to kruchy stan, wymagający ciągłego działania, do którego dążymy, ale który nie jest trwały i dany na zawsze.

2 Rozwinięcie

Hypervisor przydziela maszynom wirtualnym zasoby fizycznego serwera (pamięć RAM, przestrzeń dyskową na dane, moc obliczeniową procesorów). Odpowiada też za instalację i utrzymanie wirtualnych systemów operacyjnych (tzw. guest OS). Niestety system operacyjny wirtualnej maszyny oraz hypervisor zużywają zasoby sprzętowe fizycznego serwera. Klasyczna wirtualizacja jest bardziej zasobożerna niż konteneryzacja, co w efekcie pozostawia mniej mocy obliczeniowej dla aplikacji.

W architekturze konteneryzacji znika konieczność istnienia hypervisora oraz wirtualnych systemów operacyjnych, niejako duplikujących działanie systemu, na którym pracuje fizyczny serwer. Szybkość działania aplikacji w środowisku kontenerowym jest przez to dużo szybsza. Nie trzeba czekać na uruchomienie hypervisora i załadowanie systemu maszyny wirtualnej. Aplikacja w kontenerze startuje niemal natychmiast, a odpowiada za to jeden silnik kontenerowy, który obsługuje wszystkie kontenery w danej instancji.



Rysunek 1: Porównanie wdrożenia, wirtualizacji i konteneryzacji

Dodatkowo poszczególne kontenery stanowiące składowe większej aplikacji mogą się ze sobą bezpiecznie komunikować.

Istnieje jeszcze jedna różnica między wirtualizacją, a konteneryzacją – pierwsza z nich skupia się na wirtualizacji warstwy sprzętowej, natomiast druga odpowiada za wirtualizację platformy systemowej.

2.1 Kontenery platform chmurowych

Technologia kontenerów charakteryzuje się ciągłym wdrażaniem i testowaniem nowych rozwiązań, wysokim wykorzystaniem zasobów i ich izolacją, przez co szybko zyskała na znaczeniu w rozwoju i spopularyzowała. Jednak

w tym samym czasie bezpieczeństwo kontenerów i węzłów, opartych na kontenerach w platformach chmurowych również zyskuje na znaczeniu. Istniejące metody ochrony bezpieczeństwa mają pewne ograniczenia i nie mogą w pełni chronić bezpieczeństwa kontenerów platform chmurowych. Cechy charakterystyczne BlockChain, takie jak odporność na manipulacje, odporność na fałszowanie, dystrybucję itd., można zastosować do ochrony węzła platformy chmurowej opartej na kontenerach, znacznie poprawiając w ten sposób wydajność bezpieczeństwa.

Jego energiczny rozwój przyniósł również wiele problemów związanych z bezpieczeństwem:

- Problem z atakiem zdalnym: w wyniku zaniedbania konfiguracji portu zdalnego dostęp został otwarty, w wyniku czego złośliwy atakujący uzyskał dostęp do zdalnego kontenera. Po uruchomieniu platformy chmurowej opartej na kontenerach haker manipuluje plikiem głównym, zapisuje własny klucz prywatny ssh i ostatecznie uzyskuje zdalną kontrolę nad głównym kontenerem.

- Problem z ucieczką kontenera: gdy kontener ucieka, stwarza to ogromne zagrożenie dla wszystkich kontenerów podłączonych do węzła, jak również dla samego węzła.

- Ataki sieciowe: atakujący mogą atakować Dockera za pomocą ataków ARP, luk w zabezpieczeniach IPtable i wąchania między kontenerami.

- Ataki typu „odmowa usługi”: Jeśli w kontenerze ciągle tworzone są nieskończone ilości nowych plików, i węzeł systemu plików głównego kontenera jest wyczerpany, to w rezultacie inne kontenery nie mogą tworzyć nowych plików, a główny kontener nie może tworzyć nowych kontenerów.

W tym schemacie bezpieczne i odporne na manipulacje cechy BlockChain są zintegrowane z mechanizmem bezpieczeństwa kontenerowej platformy chmurowej. W każdym węźle tworzone jest drzewo Merkle, które łączy funkcje odporności na manipulacje i śledzenie kluczowych plików konfiguracyjnych w węzłach platformy chmurowej opartej na kontenerach, poprawiając w ten sposób wydajność zabezpieczeń węzłów. Algorytmy hashujące mogą przekształcić dane o dowolnej długości w dane o stałej długości i jest to proces nieodwracalny i niemożliwe jest odwrotne uzyskanie oryginalnych danych. Algorytmy hashujące sa również wrażliwe na zawartość danych wejściowych.

2.2 Docker

Docker jest nazwą firmy produkującej otwarte oprogramowanie o tej samej nazwie. Oprogramowanie Docker jest narzędziem ułatwiającym tworzenie, wdrażanie i uruchamianie aplikacji za pomocą kontenerów.

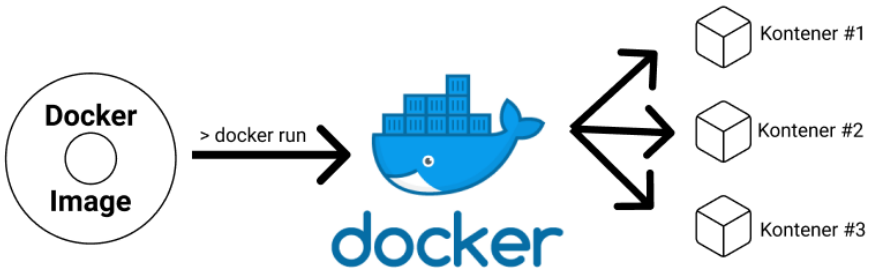
2.2.1 Docker Engine

Docker Engine jest oprogramowaniem tworzącym i uruchamiającym kontenery z pliku *Docker Image*.

2.2.2 Docker Image

Docker Image jest szablonem tylko do odczytu. Zawiera zbiór instrukcji umożliwiających stworzenie kontenera, który może być uruchomiony w ramach Dockera. Zapewnia wygodny sposób prekonfiguracji środowiska. Można go porównać do pliku obrazu systemu operacyjnego (.iso). Nowy obraz może być stworzony na dwa sposoby:

- Poprzez uruchomienie kontenera z istniejącego obrazu, dokonanie w nim zmian, a następnie zapisanie jako nowy obraz.
- Poprzez *Dockerfile* - plik tekstowy zawierający parametry tworzonego obrazu.

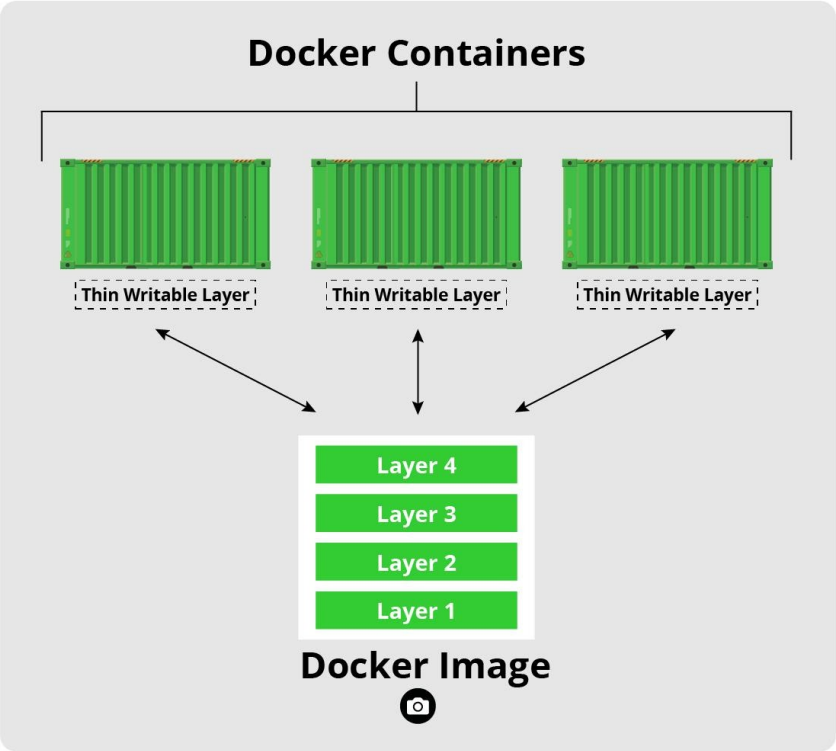


Rysunek 2: Schemat Docker Image

2.2.3 Docker layers

Każdy z plików składających się na obraz Dockera jest nazywany warstwą. Warstwy te tworzą serię pośrednich obrazów, budowanych na stosie, jeden na drugim, gdzie każda warstwa zależy od poprzedniej.

Przy każdym uruchomieniu kontenera z obrazu, Docker dodaje edytowalną warstwę znaną jako *container layer*, która przechowuje wszelkie zmiany w kontenerze w czasie swojego działania. Ta warstwa jest jedyną różnicą pomiędzy uruchomionym kontenerem, a źródłem obrazu Dockera.



Rysunek 3: Edytowalna warstwa, a Docker Image

Każda warstwa zawiera sumę kontrolną SHA256. Za jej pomocą można sprawdzić, czy obraz nie został zmodyfikowany.

2.2.4 Dlaczego warto stosować Dockera?

Docker umożliwia izolację aplikacji przy niewielkich nakładach. Dzięki warstwom, możliwe jest stworzenie wielu kontenerów, które zajmują niewiele miejsca, czego nie można powiedzieć o maszynach wirtualnych zajmujących dużo więcej miejsca.

2.2.5 Dobre praktyki bezpieczeństwa

Poniższa lista zawiera dobre praktyki bezpiecznego korzystania z platformy Docker:

- główny proces platformy należy uruchamiać na dedykowanym serwerze, odizolowanym od innych komputerów,
- najlepszą opcją jest uruchamianie głównego procesu na komputerze z systemem Unix,
- podczas konfigurowania katalogów hosta jako woluminów należy zachowywać szczególną ostrożność, ponieważ kontener może uzyskać do nich pełny dostęp i wykonywać w nich nieodwracalne operacje,
- komunikację należy zabezpieczać za pomocą protokołu SSL,
- wewnątrz kontenerów nie należy uruchamiać procesów z uprawnieniami administracyjnymi,
- warto rozważyć użycie specjalnych funkcji zabezpieczeń, np. AppArmor lub SELinux,
- kontenery, w odróżnieniu od maszyn wirtualnych, współdzielą jądro systemu hosta. Dlatego ważne jest, aby instalować najnowsze poprawki bezpieczeństwa.

Administratorzy systemu Linux powinni przestrzegać najlepszych zaleceń dotyczących bezpieczeństwa. Stosując wymienione niżej praktyki, można tworzyć bezpieczne usługi i kontenery:

- nie uruchamiaj oprogramowania jako administrator,
- blokuj uprawnienia SETUID,
- w kontenerze uruchamiaj tylko jedną aplikację lub mikro usługę,
- do współdzielenia poufnych danych nie używaj zmiennych środowiskowych ani nie uruchamiaj kontenerów w trybie uprzywilejowanym,
- sprawdzaj, jacy użytkownicy mają dostęp do hosta z platformą Docker,
- aktualizuj platformę Docker do najnowszych wersji, pozbawionych luk w bezpieczeństwie i oferujących nowe funkcjonalności,

- aktualizuj jądro systemu Linux, w którym działa platforma Docker. Współdzielone przez kontenery jądro systemu jest jednym z najbardziej newralgicznych komponentów systemu. Dlatego bardzo ważne jest, aby instalować najnowsze poprawki, gdy tylko się pojawią.
- w pliku Dockerfile można umieścić dwie instrukcje, które podpisują obraz w chwili jego kompilacji oraz publikują go w repozytorium. Dzięki temu podczas pobierania obrazu można sprawdzić plik z podpisem.

3 Podsumowanie i wnioski

Bezpieczeństwo kontenerów to nie lada wyzwanie. Jest kilka aspektów, o których należy zawsze pamiętać. Przede wszystkim aby móc uruchamiać kontenery i aplikacje, należy wcześniej uruchamiać główny proces platformy Docker, co wymaga posiadania uprawnień administracyjnych. Inną kwestią jest elastyczność platformy, dzięki której można łatwo uruchamiać wiele instancji kontenerów, z których każdy może mieć zainstalowane inne poprawki bezpieczeństwa. Platforma Docker, jak każde inne oprogramowanie, jest narażona na zagrożenia. Aby je zminimalizować, należy stosować dobre praktyki bezpieczeństwa i często weryfikować infrastrukturę pod kątem podatności na ataki.

4 Spis literatury

[1] *10 zasad bezpieczeństwa kontenerów*. https://linuxpolska.pl/blog/10-zasad-bezpieczenstwa-kontenerow/.

[2] *Bezpieczeństwo kontenerów w DevOps. Zabezpieczanie i monitorowanie kontenerów Docker*. https://helion.pl/pobierz-fragment/bekode/pdf.

[3] *Can Container Fusion Be Securely Achieved?* https://dl-1acm-1org-1rpithgax0532.han.polsl.pl/doi/10.1145/3366615.3368356.

[4] *Czym jest bezpieczeństwo kontenerów w środowisku Docker?* https://www.trendmicro.com/pl_pl/what-is/container-security/docker.html.

[5] *Konteneryzacja – czym są kontenery i skąd ich popularność?* https://fotc.com/pl/blog/konteneryzacja-czym-sa-kontenery/.

[6] *Research on the Security Protection Scheme for Container-Based Cloud Platform Node Based on BlockChain Technology*. link.springer.com/chapter/10.1007/978-981-13-2203-7_3.