

Containere – utilizări și avantaje față de mașinile virtuale

Container. Problema pe care o rezolvă

Necesitatea containerului și-a făcut simțită prezența în lumea DevOps în momentul în care a fost pusă problema rulării aplicațiilor software cu succes atunci când sunt mutate între diferite medii de lucru. În general, aplicațiile sunt dezvoltate într-un mediu de test, însă în momentul migrării acestora în producție, pot apărea probleme datorită neconcordanței între dependențe, versiuni de librării sau alte resurse folosite.

Containerele rezolvă această problemă, construind un mediu de execuție complet, cuprinzând aplicația, dependențele utilizate, librării, variabile de mediu și alte fișiere folosite de aceasta. Acestea pot fi văzute drept niște cutii ce conțin tot ceea ce are nevoie aplicația pentru a rula. Prin acest procedeu, este virtualizat sistemul de operare, iar aplicațiile se execută în medii de lucru izolate, astfel ca putem utiliza containerele oriunde, atât pe dispozitivul dezvoltatorului, cât și într-un cloud public. Deoarece crearea acestora poate fi automatizată, containerele permit o dezvoltare solidă a procesului de CI/CD.

Diferențele dintre o mașină virtuală și un container

Prin virtualizare se definește procesul de a crea un nivel intermediar de abstractizare peste sistemul hardware existent al calculatorului, cu scopul de a genera un sistem de calcul virtualizat, mai exact un calculator generat de un calculator. Aceasta abordare este utilizată de regulă în scopul execuției mai multor sisteme de operare în același calculator simultan.

Dacă un container realizează abstractizarea sistemelor de operare, permițând execuția aplicațiilor, în cazul mașinilor virtuale virtualizarea are loc la nivelul hardware-ului. Astfel, putem vedea o mașină virtuală ca un mediu de lucru simulat pe calculatorul gazdă.

Virtualizarea fiecărei componente hardware care compune mașina gazdă (procesoare, RAM, etc.) este realizată prin intermediul unui software numit hypervisor. El reprezintă un layer intermediar între zona hardware fizică și mediile de lucru virtualizate și facilitează funcționarea acestora în mod simultan prin preluarea resurselor din infrastructură și redirectionarea acestora către instanțele create.

În multe situații, containerele reprezintă o alternativă mai avantajoasă față de mașinile virtuale. În primul rând mașinile virtuale conțin un sistem de operare și un kernel propriu, împreună o copie virtuală a hardware-ului, spre deosebire de containere, care virtualizează doar sistemul de operare, în timp ce folosesc kernelul host-ului. Astfel că o variantă mai ușoară, rapidă, eficientă, ce necesită mai puțină memorie pe mașina gazdă și oferă aceleași funcționalități ca mașinile virtuale poate fi considerată un container. De asemenea, containerele pot fi considerate o soluție viabilă în dezvoltarea unei aplicații bazate pe microservicii.

Docker. Utilizări

Docker este o platformă open source ce poate fi utilizată pentru a administra aplicații într-un mediu de lucru virtualizat, sub formă de containere. Dat fiind caracterul izolat pe care îl aduc containerele, utilizarea și procesul de deployment al aplicațiilor devine mai ușoară, iar dezvoltatorii software nu vor întâlni diferite probleme de compatibilitate.

În ceea ce privește arhitectura Docker, acesta are la bază mai multe componente cheie. *Docker Engine* reprezintă nucleul sistemului Docker, acesta ocupându-se de crearea, execuția și administrarea containerelor. În componența acestuia se află *Docker daemon*, serverul care se ocupă de operațiile amintite anterior, *Docker CLI* care permite utilizatorului să trimită comenzi Docker și *Rest API*-ul ce face legătura dintre Docker și programe.

Dockerfile este un script specializat ce conține instrucțiunile de creare a imaginii Docker, precum imaginea sistemul de operare utilizat (specificat prin instrucțiunea *from*), variabile de mediu (indicate prin *env key value*), fișiere și alte resurse de care e nevoie pentru a rula imaginea.

O *image de Docker* este un pachet read-only ce conține codul sursă, dependențele, librăriile și alte resurse necesare pentru crearea imaginii dorite. Aceasta este imutabilă și este construită de regulă pornind de la o imagine de bază, asupra căreia se adaugă diverse elemente în funcție de caz.

Docker registry poate fi văzut ca un registru de imagini Docker pe care le putem folosi în dezvoltarea aplicațiilor. Acesta se întindește sub două forme, on-premise și în cloud (ca Docker Hub). În funcție de Docker registry-ul configurat local, în momentul execuției comenzilor *docker pull* sau *docker run* imaginile cerute sunt extrase din registry, iar comanda *docker push* încarcă imaginea în registry.

Alte instrumente dedicate containerelor

Linux Containers (LXC) a fost prima tehnologie de administrare a containerelor, ce permite execuția simultană a mai multor sisteme Linux izolate pe aceeași gazdă.

Kubernetes este un orchestrator de containere, ce ajută la automatizarea administrării acestora, în special în cazul aplicațiilor bazate pe microservicii, care utilizează sute sau mii de containerere.

Bibliografie

1. Molly Clancy, 15 octombrie 2021, „What’s the Diff: VMs vs. Containers”
<https://www.backblaze.com/blog/vm-vs-containers/>
2. IBM Cloud Team, IBM Cloud, 9 aprilie 2021, „Containers vs. Virtual Machines (VMs): What’s the Difference?”
<https://www.ibm.com/cloud/blog/containers-vs-vm>

3. Google Cloud Documentation, „What are Containers?”
<https://cloud.google.com/learn/what-are-containers>
4. David Zomaya, 21 octombrie 2019, „Container vs. Hypervisor: What’s the Difference?”
<https://www.cbttuggets.com/blog/certifications/cloud/container-v-hypervisor-whats-the-difference>
5. Sofija Simic, 16 septembrie 2021, „What is Docker?”
<https://phoenixnap.com/kb/what-is-docker>