

DOCKER KONTEINERU IZMANTOŠANAS IESPĒJAS POSSIBILITIES OF USING DOCKER CONTAINERS

Autors: **Dainis ŽOGOTS**, e-pasts: dainiszogots2@inbox.lv
Zinātniskā darba vadītāja: **Sandra EŽMALE**, Dr.oec, e-pasts: Sandra.Ezmale@rta.lv

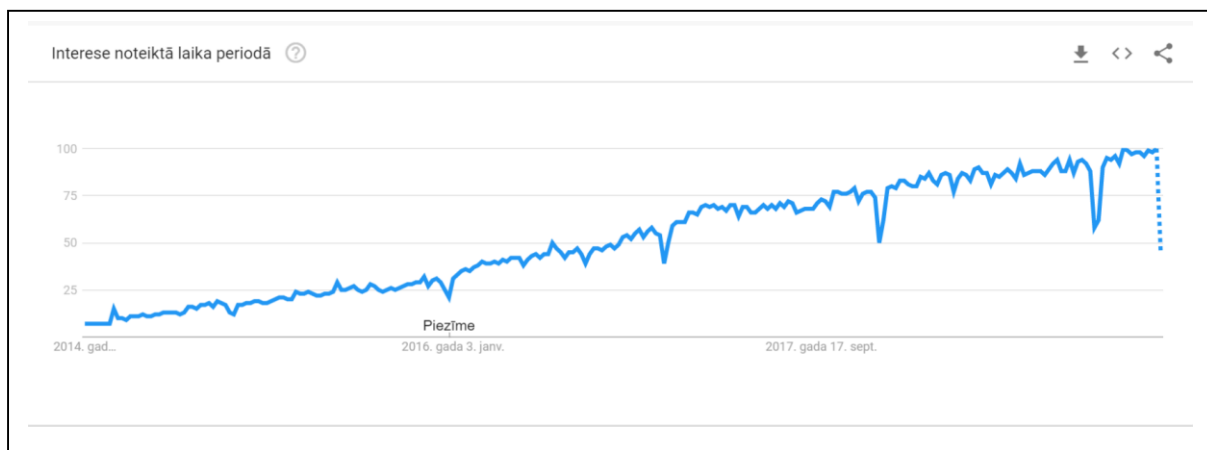
Abstract. *Considering that in nowadays virtualization in various Information Technology (IT) systems is becoming more demanding, it is necessary to deeply understand the possibilities, advantages and risks of that kind of technologies. One of these virtualization tools is the Docker containers, which are becoming more convenient, faster, more popular and more profitable than virtual machines (VM). The aim of this work is to explore and analyze the possibilities of Docker containers and to evaluate the pros and cons of the given technology. Also determine when it is worth using Docker containers as virtualization tools for an IT system.*

Keywords: *Deployment, Docker, Container, Virtualization, Virtual Machine.*

Ievads

Mūsdienās IT sistēmas izstrādes procesā ļoti liela loma ir virtualizācijai, kas parasti ir nepieciešama priekš tā, lai izolētu kādas IT sistēmas moduļus, kas uzlabotu kopējo kāda produkta darbību un stabilitāti, kura ir ļoti svarīga katram veiksmīgam IT produktam. Agrāk priekš tā, lai izolētu IT sistēmas daļas tika izmantotas virtuālās mašīnas (VM), kuras savā laikā gan arī tagad ir diezgan pieprasītas un arī tiek izmantotas. Taču dotajai pieejai ir viens liels mīnuss, ka uz katras virtuālās mašīnas ir nepieciešams uzstādīt savu operētājsistēmu (OS), kas nozīmē, ka ja uz viena servera ir uzstādītas vairākas virtuālās mašīnas, tad diezgan liela resursu daļa tiks izmantota nevis uz produkta darbību, bet gan uz katras operētājsistēmas darbību. [1] Tāpēc, kā viens no veidiem kā risināt doto problēmu un ietaupīt resursus ir izmantot *Docker* konteinerus.

Dotais temats šobrīd kļūst aizvien aktuālāks (sk. 1.att.), jo ikvienai populārajai un stabili IT sistēmai ir ļoti svarīgi, lai viņu sistēma būtu vēl stabilāka un pieejama jebkurā laikā, kā arī, lai dažādās kritiskās situācijās būtu iespēja visu izlabot pēc iespējas ātrāk un kvalitatīvāk, kas uzlabotu gan lietotāju pieredzi, gan IT sistēmas stabilitāti. Ļoti būtiski vēl ir tas, ka dotā tehnoloģija ļauj samazināt izmaksas uz serveru komponentēm un to uzturēšanu, kas laika gaitā var izveidot diezgan lielus ietaupījumus.



1.attēls. *Docker* popularitātes pieaugums periodā (2014 - 2019) gadam [2]

Mūsdienās serveru virtualizācijas pielietojums kļūst aizvien pieprasītāks un populārāks, kas ir saistīts ar to, kādus ieguvumus virtualizācija spēj sniegt. Dotā darba mērķis ir izpētīt un

noteikt *Docker* konteineru izmantošanas iespējas, kādi ir to plusi un mīnusi, kā arī apskatīt to atšķirību no virtuālajām mašīnām (VM).

Materiāli un metodes **Serveru virtualizācija**

Pēc būtības serveru virtualizācija ir process, kurā uz kādas fiziskas mašīnas (servera) tiek izmantota programmatūra priekš tā, lai izveidotu vairākas izolētas virtuālas mašīnas (VM). Kur savukārt galvenajai fiziskajai mašīnai ir viena Operētājsistēma un tā var uzturēt vairākas virtuālas mašīnas ar dažādam Operētājsistēmām. Dotā pieeja ir ļoti izdevīga, ja ir nepieciešamība izmantot vairākas informāciju tehnoloģiju sistēmas vai arī vairākas operētājsistēmas, lai samazinātu izmaksas uz elektroenerģiju, servera komponentēm un speciālām telpu nomām. [3]

Kā galvenie iemesli kāpēc tiek izmantota serveru virtualizācija ir:

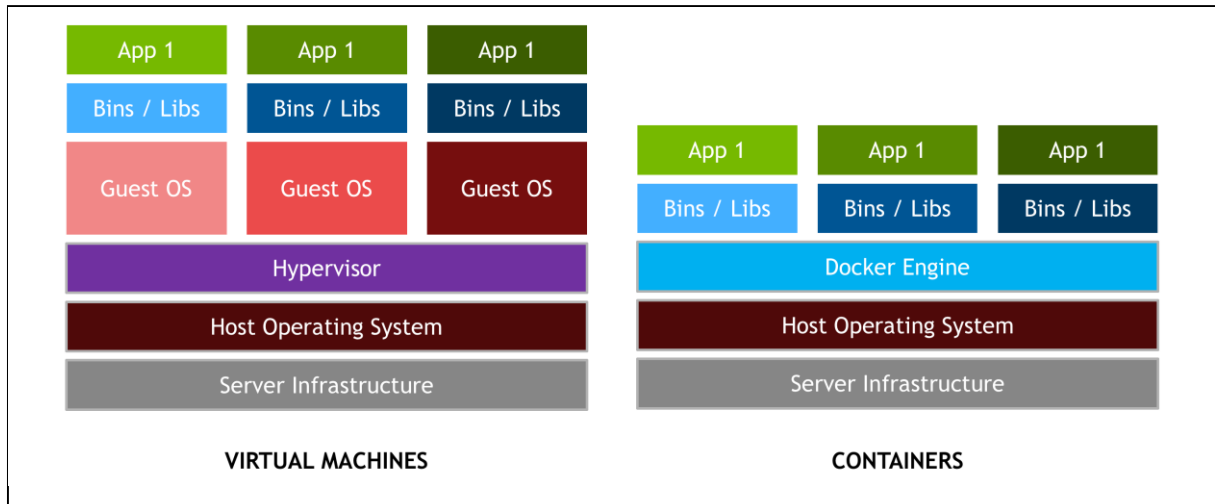
- Serveru administrēšanas centralizācija;
- Uzlabot serveru pieejamību;
- Informācijas sistēmas vieglāka atjaunošana, pēc sistēmas kritiskas apstāšanās;
- Programmatūras izstrādes un testēšanas atvieglošana;
- Galvenā servera resursu sabalansētāka izmantošana. [4].

***Docker* konteineri**

Docker konteineri pirmo reizi parādījās 2013. gadā, kā atvērtā koda projekts, kurš jau uz to brīdi izmantoja *Linux* operētājsistēmā eksistējošus jēdzienus kā *cgroups* un *namespaces*. Dotā tehnoloģija ir unikāla, jo tā fokusējas uz izstrādātāju prasībām, lai atdalītu kādas informācijas sistēmas atkarības no sistēmas infrastruktūras. [5]

Docker konteineris ir standarta programmatūra, kas sevī satur kādas programmas kodu un visas nepieciešamās programmas bibliotēkas, lai programmproduktu varētu ātri un droši palaist uz citām skaitļošanas vidēm. [5] Konteineri, atšķirībā no virtuālajām mašīnām nodrošina augstāku veiktspēju, jo nav tiem nav nepieciešami papildus resursi katras OS darbināšanai. (sk. 2. att.) Tostarp, lai uz kāda servera varētu lietot konteinerus ir nepieciešams uzstādīt *Docker Engine*, kas ir galvenais dotās programmatūras dzinējs un atbild par konteineru darbināšanu, un pārvaldīšanu.

Līdzīgi kā virtuālās mašīnas arī konteineri darbojas drošā un izolētā vidē. Konteineri ir mazāki un ātrāki nekā virtuālās mašīnas, kas nosaka to, ka to izveide, migrēšana un palaišana ir daudz reizes ātrāka, kas ir ļoti svarīgs aspekts programmprodukta izstrādes un uzturēšanas procesā. Papildus tam, mazais konteineru izmērs ļauj uz vienas fiziskās mašīnas izveidot daudz vairāk konteineru nekā virtuālo mašīnu.



2.attēls. Atšķirība starp virtuālo mašīnu un *Docker* konteineri [6]
Docker konteineru izmantošanas iespējas

Visbiežāk *Docker* konteineri tiek izmantoti sekojošos veidos:

- Eksistējošu programmu migrēšana uz mūsdienīgāku vidi – dažas organizācijas izmanto *Docker* konteinerus priekš tā, lai migrētu esošus IT produktus uz virtualizācijas vidi. Kaut arī dotā iespēja sevī ietver operētājsistēmas virtualizācijas priekšrocības, tā nepiedāvā pilnīgu programmatūras moduļu, konteineru arhitektūras priekšrocības;
- Programmprodukta pirmkoda pārstrādes analīze priekš konteinerizācijas – lai gan pirmkoda pārstrāde ir daudz laikietilpīgāka nekā eksistējošu produktu migrēšana, tā ļauj pilnībā izmantot konteineru vidi;
- Izveidot programmproduktus pilnībā no jauna ar konteinerizācijas atbalstu – tā pat kā pirmkoda pārstrāde, dotais variants ļauj pilnībā izmantot konteineru vidi;
- Nodrošināt labu mikro servisu arhitektūru – sadalītas programmas un mikro servisi var būt daudz vienkāršāk izolēti, izvietoti publiskā pieejā un pielāgoti izmantojot atsevišķus konteineru blokus;
- Nodrošināt *DevOps* atbalstu nepārtrauktai integrācijai un produkta izvietojumam – *Docker* konteineru tehnoloģija ļauj sagatavot, testēt un izvietot programmproduktu no viena un tā paša konteineru šablona (*Image*);
- Nodrošināt atkārtotu darbu (*Jobs*) un uzdevumu (*Tasks*) izpildi un izvietojumu – konteinerus izmanto priekš tā, lai darbinātu dažādus procesus fona režīmā par kuriem parasts lietotājs pat neiedomājas. [7]

***Docker* konteineru plusi un mīnusi**

Kaut arī *Docker* konteineri ir kļuvuši ļoti populāri un tos izmanto aizvien vairāk informācijas tehnoloģiju produktu izstrādē, tam ir savi plusi un mīnusi. No plusiem konteineriem ir”

- Konteineri ir mazi – salīdzinājumā ar virtuālajām mašīnām, kuru izmērs sākas ar gigabaitiem, tad konteineru izmērs sākas ar 10 megabaitiem, kas ievērojami ietaupa resursus;
- Izmanto mazāk servera resursus;
- Spēja ļoti ātri startēt – parasti aizņem dažas sekundes, kas ir ļoti lietderīgi, kad ir nepieciešamība restartēt vai arī uzstādīt atjauninājumu programmproduktam;

- Strādā uz jebkuras mašīnas (datora) – dotā situācija ir ļoti lietderīga, kad iet runa par darbu komandā, jo konteineri nodrošina to, ka viens un tas pats produkts strādās uz katra datora identiski;
- *DevOps* atbalsts – konteineru virtualizācija ir lielisks rīks priekš mikro servisu arhitektūras un *DevOps*. [8]

Lai gan *Docker* konteineru plusu ir diezgan daudz, taču arī ir lietas par kurām noteikti ir jāpārzina un jāseko līdzi (mīnusi):

- Konteineru drošība – Tā kā konteineros nav pilnīgas operētājsistēmas, tas nozīmē, ka konteineri, kuri nav aizsargāti korekti, ļoti bieži kļūst par vienkāršu mērķi hakeriem;
- Izolācija – Tā kā konteineri izmanto vienu un to pašu *Kernel* (datorprogramma), tas nozīmē, ka tie nav izolēti uz visiem 100%, tāpēc ir jāizvairās no vairāku konteineru izmantošanas uz viena servera vai arī ir nepieciešams ļoti labi pārzināt to, kā organizēt konteineru darbību uz viena servera;
- Tīklošana – Dotā iespēja ir ļoti noderīga, taču tā var būt diezgan sarežģīta konfigurēšanas procesā. [8]

Rezultāti un to izvērtējums

Docker konteineri no parādīšanās brīža kļūst aizvien populārāka tehnoloģija, kuras potenciāls kļūst aizvien lielāks. Tāpēc ir jāseko dotās tehnoloģijas attīstības gaitai, jo tieši šī tehnoloģija ļauj sākt ekonomēt uz viena no izdevumiem kādas informācijas sistēmas uzturēšanā. No dotās tehnoloģijas ieguvēji ir gan programmaprodukta izstrādātāji, gan pats uzņēmums, gan arī informācijas sistēmas gala lietotāji. Neskatoties uz to, ka *Docker* konteineri kļūst aizvien populārāki dažos gadījumos tomēr ir labāk izmantot virtuālās mašīnas (VM). Lai gan konteineriem ir ļoti daudz priekšrocību izmantošanā, vienmēr ir jāņem vērā, ka priekš efektīvas to izmantošanas ir nepieciešamas padziļinātas zināšanas tīklošanā, konteineru drošībā un tajā, kā un kad ir labāk tos izmantot, jo dažos gadījumos tie var būt nevajadzīgi, un neefektīvi.

Secinājumi

Kā jebkurai IT tehnoloģijai, kura aktīvi attīstās un, kuras popularitāte aug ir svarīgi sekot tās attīstībai, jo ļoti bieži gadās tā, ka atjauninājumi kādai tehnoloģijai var izmainīt tekošās sistēmas darbību. Ļoti svarīgs aspekts, ko ir jāievēro, kad izmanto *Docker* konteinerus ir to drošība, kas mūsdienās ir ļoti svarīgi, it īpaši, kad informācijas sistēma sevī satur lietotāju datus. Savukārt viena no galvenajām konteineru priekšrocībām ir tāda, ka pateicoties tai ir iespēja izveidot mikro servisu arhitektūru, kura ļauj izveidot modulāru informācijas sistēmu. Kā arī *Docker* konteineru izmantošana samazina kopējās izmaksas par produkta izvietošanu publiskā pieejā, jo priekš to darbināšanas ir nepieciešams mazāk servera resursu nekā tas būtu vajadzīgs priekš virtuālās mašīnas. Turpinoties informācijas sistēmu apjomu pieaugumam, pieprasījums pēc *Docker* konteineriem augs un to efektivitāte kļūs aizvien lielāka. Darba autors secina, ka mūsdienās ir ļoti lietderīgi un efektīvi izmantot *Docker* konteinerus, taču ir nepieciešamība ļoti labi pārzināt to izmantošanu.

Summary

In this paper is described the main differences between Docker containers and virtual machines (VMs) and describes how to use Docker containers and what pros and cons does Docker containers have.

The aim of this work is to explore and describe the use cases of Docker containers, their pros and cons, and what is the difference from virtual machines (VMs).

In the past, software developers used virtual machines (VMs) to develop information systems and isolate their modules, but their size, resource usage and speed are often insufficient. So in nowadays software developers as well as various companies use Docker containers, which take up much less space on the server, use less server resources and is more efficient when it comes to create a new isolated environment where it supports to deliver the final product faster, test them and manage each module of system.

As an example, many large companies today use containers architecture, such as Google, where everything like Gmail, Google Search and YouTube runs in containers.

Unlike virtual machines, Docker containers do not require a Guest Operating System, which significantly saves the size of the server. For example, the size of virtual machine usually starts from gigabytes, while the size of the containers starts from 10 megabytes, which makes the containers much faster for launch and change implementation.

The main advantages of Docker containers are small size, use fewer server resources, have ability to start in a few seconds, can run on any other computer or server and supports microservice architecture.

From cons, Docker containers include: Docker networking configuration, Docker container security configuration and container isolation on one server.

In nowadays, many people are trying to move to Docker containers architecture, which indicates that there is a great prospect in future for containerization and knowledge of Docker containers will greatly improve the performance, stability and quality of the any information system.

Literatūra

1. Habr. Docker. начало. Sk. Internetā (17.04.2019) <https://habr.com/ru/post/353238/>
2. Docker trend. Sk. Internetā (17.04.2019) <https://trends.google.com/trends/explore?date=today%205-y&q=%2Fm%2F0wkcjgj>
3. Cisco. What is Server Virtualization in Networking? Sk. Internetā (18.04.2019) <https://www.cisco.com/c/en/us/solutions/small-business/resource-center/tools-tips/virtualization.html>
4. W3schools. Server Virtualization. Sk. Internetā. (18.04.2019) <https://www.w3schools.in/cloud-virtualization/server-virtualization/>
5. Docker. What is Container? Sk. Internetā. (18.04.2019) <https://www.docker.com/resources/what-container>
6. Nvidia DevBlogs. VM vs Docker. Sk. Internetā. (18.04.2019) https://devblogs.nvidia.com/wp-content/uploads/2016/06/VM_vs_Docker.png
7. NetApp. What are Container? Sk. Internetā (18.04.2019) <https://www.netapp.com/us/info/what-are-containers.aspx>
8. Koukia. Why Docker? Pros and Cons. Sk. Internetā (19.04.2019) <https://koukia.ca/why-docker-pros-and-cons-949d104478c5>