KAUNO TECHNOLOGIJOS UNIVERSITETAS

KOMPIUTERIŲ KATEDRA

Virtualios infrastruktūros sauga

T120M144

Laboratorinio darbo Nr. 1 ataskaita

Konteineriai. Docker

Atliko: gr.

Stud.

Patikrino:

Kaunas, 2022

# Laboratorinis darbas Nr. 1.

**WEB paslaugos paleidimas naudojant Docker tipo konteinerius**

## Darbo tikslas

Apjungiant docker konteinerius į vieną aplinką, paleisti WEB (tinklapių) paslaugą (service) su duomenų baze.

## Darbo priemonės

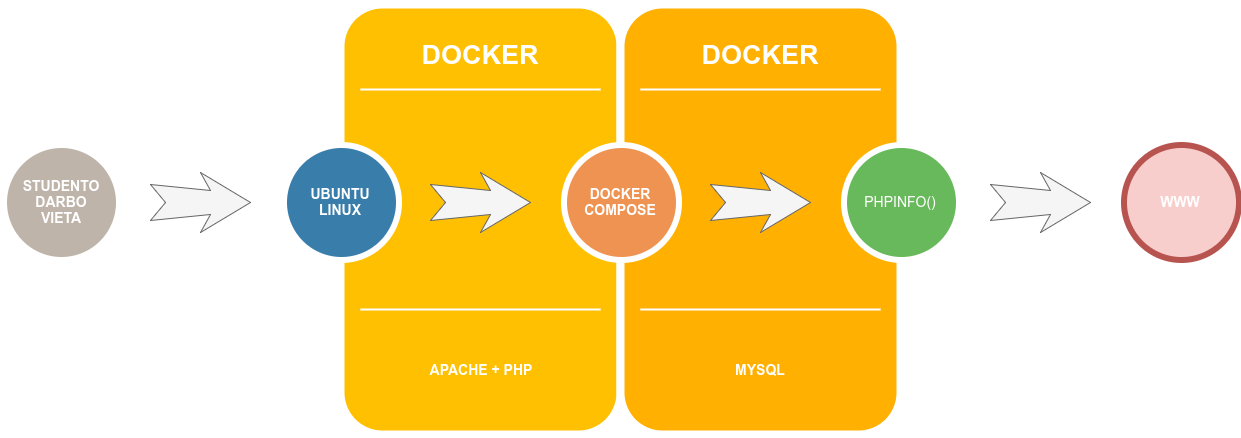
1. Linux Ubuntu 17 Vmware atvaizdas (Paleidimui naudojama VirtualBox aplinka. Ją galima parsisiųsti nemokamai iš <https://www.virtualbox.org/>
2. Docker aplinka (įdiegta). Jei tai atliekama iš namų ir naudojama Windows OS, galima Docker diegti lokaliai (<https://docs.docker.com/desktop/windows/install/>)
3. docker-compose (įdiegta). Jei tai atliekama iš namų ir naudojama Windows OS, galima docker-compose diegti lokaliai (<https://docs.docker.com/compose/install/>).
4. Prieiga (naudotojas yra „sudo“ grupėje): naudotojas **stud** slaptažodis **stud**
5. FTP prieiga (Linux Ubuntu atvaizdas – VIS katalogas). Prisijungimo vardas: anonymous (Reikalinga naudoti KTU VPN).
6. Galima naudoti <https://www.docker.com/> tinklapyje esančią informacija ir Docker virtualizacija įsidiegti lokalioje darbo vietoje (tie kas dirba nuotoliniu būdu). Diegimas Windows aplinkoje - <https://docs.docker.com/docker-for-windows/install/> Diegimas Apple OS aplinkoje - <https://docs.docker.com/docker-for-mac/>

## Darbo uždaviniai

1. Naudojant docker konteinerius sukurti web paslaugai tinkamą aplinką (http, php, mysql – privalomai)
2. Sukurti ir aprašyti docker konteinerių veikimą docker-compose.yml faile.
3. Gebėti pademonstruoti phpinfo funkciją, naudojant jau sukurta web aplinka.
4. Pateikti darbo rezultatų išvadas.

## Darbo planas

1. Pasileisti Ubuntu Linux aplinką (jei nėra įdiegtas Docker lokaliai)
2. Išsirinkti (savarankiškai apsispręsti geriausią sprendimą tinkanti Jus) docker konteinerius, kurie bus diegiami į virtualią aplinką.
3. Aprašyti docker konteinerius
4. Patikrinti ar po perkrovimo („reboot“) sistema tinkamai startuoja.
5. Sukurti index.php failą su phpinfo funkcija ir ji patalpinti tinkamoje vietoje, kad jį pasiekti iš *host* mašinos (žr. paveikslą).
6. Pateikti išvadas ir docker-compose failą



**Darbo eiga**

1. Įdiegiame Linux’ine mašiną (atlikta).

2. Prisijungiame kaip "root" naudotojas

2.1. **sudo -i**

2.2. Įvedame "stud" naudotojo slaptažodį

3. parsisiunčiam konteinerį apache+php5- nimmis/apache-php5

4. parsisiunčiam konteinerį mysql

5. Patikriname ar repozitorija sukurta: **docker images**

6. Sukrutį failą docker-compose.yml: **cd /opt/**

6.1. vim docker-compose.yml

6.2. Įterpti eilutes tokias, kad atliktų sekantį funkcionalumą:

prievadas 80 - iš *host* mašinos turi būti pasiekiamas

Duomenų bazės informacija

MYSQL\_PORT\_3306\_TCP: 3306

MYSQL\_DATABASE: STUDENTO\_NR

MYSQL\_ROOT\_PASSWORD: NERAPASS123

6.3. Išsaugoti failą

7. bandome paleisti konteinerius būdami /opt/ direktorijoje (cd /opt/): **docker-compose up -d**

8. žiūrime ar konteineriai paleisti: **docker ps -a**

9. Bandome prisijungti prie paleisto konteinerio "pirmas" : **docker exec -it pirmas bash** ( "pirmas"<- Vardą, kurį galime rasti parašius komandą "docker ps -a")

9.1. išeiti iš konteinerio : **exit**

10. Bandome prisijungti prie paleisto konteinerio "antras\_db" : **docker exec -it antras\_db bash** ( "antras\_db" <- Vardą, kurį galime rasti parašius komandą "docker ps -a")

10.1. Prisijungti prie duomenų bazės: **mysql -u root -pNERAPASS123**

10.1.1. Žiūrėti kokios lentelės: **SHOW DATABASES;** <-- komanda, jeigu reikia.

10.1.2. Sukurti naują lentelę: **CREATE DATABASE studento\_nr;**

10.1.3. Išeiti: **exit**

10.2. Išeiti iš konteinerio : **exit**

11. Pažiūrėti kokį IP adresą gavo konteineris: **docker inspect pirmas | grep IPAddress**, **docker inspect pirmas | grep IPAddress**

12. Paleidžiame naršyklę *host’o* mašinoje ir pažiūrime ar veikia mūsų web'as.

13. Įdiegti mysql klientą pagrindinėje mašinoje: **apt-get install mysql-client**

14. Einame į www direktoriją būdami savo pagrindinėje mašinoje, o ne konteineryje: **cd /var/www/htmls**

15. Sukuriam index.php su phpinfo() funkcija ir patalpinam **/var/www/htmls**.

16. Patikrinam ar yra išvedama visa informacija apie httpd paslaugą. Kartojame 12 punktą.

**Pademonstruoti**

1. Įdiegtų docker kiekį

(darbalaukio vaizdai, kur matomas docker kiekis, išvardinti docker konteinerius)

1 lentelė

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Eil. Nr. | docker pavadinimas | docker mašinos namų katalogas | docker failas | docker versija | Pastabos\* |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

\*PASTABA: kita jūsų nuomonė svarbi informacija apie docker konteinerį (neprivaloma)

1. docker-compose nustatymai ir pasirinkimai

(darbalaukio vaizdai: nustatymo vaizdai / įkopijuoti konfigūracijos nustatymai)

1. phpinfo() informacija

(darbalaukio vaizdai: nustatymo vaizdai )

**Išvados**

(pateikiamos studento pastabos atliekant docker sujungimą)

**Darbo rezultatų vertinimas**

(pildo dėstytojas)

2 lentelė

|  |  |  |  |
| --- | --- | --- | --- |
| Vertinimas  (balais) | Galimas maksimalus vertinimo balas | Vertinimo objektas | Pastabos |
|  | 5 | Sukurti ir sujungtos į bendrą potinklį docker konteineriai | Privalomai, bent du konteineriai |
|  | 3 | Paleisti web paslauga | Apache+PHP+Mysql |
|  | 2 | docker-compose | Pateiktas docker-compose failas be klaidų |
|  | 10 |  | |