



Laborator 07

Scopul acestui laborator este explorarea lucrului cu Docker.

Atenție la copy-paste unele simboluri gen “-“ se copiază greșit.

Exerciții























1. Treceți prin primele 4 scenarii Docker de pe <http://katacoda.com/courses/docker>
2. Reminder lab trecut: Faceți un cont de student pe [Azure](#).
 - o Se va folosi contul de e-mail mta.ro. (Pe alt cont **NU** se poate activa versiunea pentru studenți).
 - o Parola de la e-mail e diferită de cea pentru wiki.
 - o Acesta nu necesită card. Astfel, nu sunteți expus unui risc de cost.
 - o Aveți la dispoziție 100\$.
3. Se vor porni 2 mașini virtuale Linux.

- o Se va extinde meniul. 

- o Se va selecta Virtual Machines.  Virtual machines

- o Se va adăuga Mașină.  Add 

- o Se vor crea 2 mașini cu următoarele caracteristici:

Subscription *		Azure for Students	
Resource group *		(New) test	
		Create new	
Instance details			
Virtual machine name *		server1	
Region *		(Europe) Germany West Central	
Availability options		No infrastructure redundancy required	
Image *		Ubuntu Server 18.04 LTS - Gen1	
		See all images	
Azure Spot instance		<input type="checkbox"/>	
Size *		Standard_B1s - 1 vcpu, 1 GiB memory (€7.39/month)	
		See all sizes	
Administrator account			
Authentication type		<input type="radio"/> SSH public key	
		<input checked="" type="radio"/> Password	
Username *		student	
Password *		••••••••••	
Confirm password *		••••••••••	



- **ATENȚIE de la *Select inbound ports* dorim și SSH și HTTP de această dată.**
 - Se va merge la Review+Create și se va apăsa Create.
 - Nu uitați, vrem 2 astfel de mașini. Cealaltă se va numi Server2.
4. Prin putty conectați-vă la cele 2 mașini (un terminal putty pentru fiecare).
- IP-ul îl puteți găsi în pagina de informații a mașinii.
 - Veți folosi student@IP_public dar notați și IP-ul privat al celor 2 mașini.

Orice greșeală faceți legată de container puteți să vizualizați containerele și să le ștergeți folosind:

- `docker container ls -a`
- `docker container rm -f CONTAINER_NAME`

Deocamdată vom lucra doar pe Server1:

5. Va trebui să copiați fișierele din schelet din folderul server1 pe server1. Puteți folosi WinScp.
6. În arhiva de laborator aveți fișierele .php pentru un site web. Acest site are nevoie de mysql si php. **Pentru acest exercițiu vă recomandăm să vă folosiți excesiv de internet.**
- Serverul mysql va avea user root și parola student
 - Instalați servere mysql, apache2, php, suport mysql pentru php. Pentru instalare puteți folosi: `sudo apt-get install SERVER_NAME`
 - În serverul mysql trebuie să introduceți baza de date din fișierul sql din folderul server1. Pentru serverul web va trebui să introduceți în folderul specific fișierul index.php din folderul server1.
 - **Verificare:** Se deschide din browser http://IP_PUBLIC_SERVER1/
Ar trebui să vedeți ceva similar (evident fără vre-o eroare):
Hello, 127.0.0.1, you accessed the page 8 times.
 - Dacă site-ul nu afișează mesajul de "Hello" este posibil să fie necesare unele [reinstalări](#). Dezinstalați/Instalați până ce site-ul funcționează. (Este posibil ca acest exercițiu să dureze mai mult. Don't give up easily.)
 - Pentru setare parolă mysql:

```
sudo mysql
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
BY 'student';
FLUSH PRIVILEGES;
exit
```

Vrem să instalăm același site web pe Server2 folosind Docker:

7. Instalați pe Server2 Docker.
- Sunt necesare mai multe comenzi complexe, acestea le găsiți în fișierul **installDocker.sh**. Pot fi probleme la rulare, executați comandă cu comandă.



- **Verificare:** `docker`
- 8. Va trebui să copiați fișierele din schelet din folderul server2 pe Server2. Puteți folosi WinScp.
- 9. Vom avea nevoie de o rețea virtuală specială pentru aceste containere:
 - Având o astfel de rețea putem adresa containerele folosind numele lor în loc de IP-uri aceasta având suport DNS built-in.
 - `docker network create MY_NETWORK_NAME`
 - **Verificare:** `docker network ls`
- 10. Aplicația care va rula pe serverul web necesită o bază de date **mysql**:
 - Acesta este un server des folosit și astfel are o imagine pe docker hub.
 - Rulați:
`docker container run -e MYSQL_ROOT_PASSWORD=student --network MY_NETWORK_NAME --name mysql-server -d mysql:5.7`
 - 1. Containerul mysql va trebui să fie în aceeași rețea virtuală cu containerul apache.
 - 2. Este important ca acest container să aibă numele **mysql-server**. Acest nume este folosit în fișierul index.php. Îl puteți modifica doar dacă îl modificați peste tot.
 - 3. În final -d setează containerul să ruleze ca un daemon, în background iar mysql este imaginea luată de pe dockerhub.
 - Adăugați baza de date în mysql:
`docker exec -i mysql-server sh -c 'exec mysql -uroot -p"$MYSQL_ROOT_PASSWORD"' < server2/all-databases.sql`
 - **Verificare:** `docker container exec -it mysql-server /bin/bash` O dată intrați în container puteți da `mysql` și apoi comenzi `mysql gen show databases;` pentru listarea tuturor bazelor de date. Din aplicația `mysql` și din container se poate ieși folosind `exit`.
- 11. Construcția imaginea pentru serverul web:
 - Acest container este unul particularizat pentru acest laborator, astfel imaginea pentru el va trebui construită din Dockerfile. Cod-ul ce va rula pe acesta este în index.php.
 - Analizați fișierul **Dockerfile**.
 - 1. Acesta pornește de la imaginea php pentru apache de pe dockerhub.
 - 2. Se instalează suport php pentru a putea comunica cu un server mysql.
 - 3. Se copiază fișierul index.php în /var/www/html/. Acesta va fi rulat când accesăm serverul.
 - Construiți imaginea: `docker build -t IMAGE_TAG_NAME`. **NU uitați de punct, aceasta este calea către Dockerfile.**
 - **Verificare:** `docker image ls`.
- 12. Rulare container server web:
 - Puteți porni de la comanda: `docker container run IMAGE_TAG_NAME`
 - Dacă pur și simplu ați dat comanda de mai sus când veți lista containerele veți descoperii că acesta deja s-a oprit. Vedeți la începutul laboratorului cum puteți șterge containerul creat.
 - [Bazat pe documentație](#) comenzii de mai sus trebuie să adăugați următoarele:



1. Conexiune la rețeaua creată la punctul 9. (Necesită și parametru)
 2. Rularea containerului în background.
 3. Asignarea unui nume containerului. (Necesită și parametru)
 4. Publicarea portului http (-p 80:80)
- **Verificare:** Din browser intrați pe adresa publică a server2. Va apărea similar:

Hello, 127.0.0.1, you accessed the page 8 times.

Laboratorul va fi prezentat. Veți intra pe rând pe teams. Va trebui să aveți terminal putty deschis la mașina principală și să dați docker container ls. De asemenea va trebui să aveți deschis site-ul Azure și site-ul din server1 și din server2. **Toate setările ar trebui să le faceți dinainte.**

După prezentarea laboratorului mergeți pe Azure în tab-ul Resources și ștergeți toate resursele create.