



Laborator 08

Scopul acestui laborator este crearea unui mini-cluster Docker Swarm și rularea unui site web pe mai multe mașini cu suport load balance.

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

Exerciții

1. Treceți prin primele 2 scenarii despre Containere de pe:
<https://katacoda.com/courses/container-runtimes>
2. Treceți prin tutorialul de Docker Swarm de pe:
<https://katacoda.com/courses/docker/getting-started-with-swarm-mode>
3. Se vor porni 3 mașini virtuale Linux.

- Se va extinde meniul.



- Se va selecta Virtual Machines.



Virtual machines

- Se va adăuga Mașină.

+ Add ▾

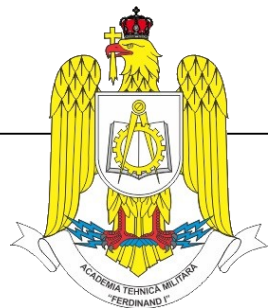
- Se vor crea 3 mașini cu următoarele caracteristici:

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 *tab-ul de Networking* selectați NIC network security group None.**



- Se va merge la Review+Create și se va apăsa Create.
 - Nu uitați, vrem 3 astfel de mașini. Celelalte se vor numi server2 și server3.
4. Prin putty conectați-vă la cele 3 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 3 mașini. **E important ca toate să se afle în aceeași locație, și în același grup.**
5. Testați conexiunea între cele 3 mașini.
- Pe una se va porni `nc -l -p 5000`
 - Pe celelalte două se va porni `nc IP_PRIVAT_PRIMA 5000`
 - Trebuie să se poată apoi scrie mesaje ce ajung de la o mașină la alta (ca un chat).
 - Testul se va face separat pentru fiecare pereche de mașini.
 - Dacă aveți nevoie de root puteți scrie `sudo su`
6. Instalați pe toate 3 mașinile 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`
7. Porniți și setați docker swarm:
- Pe server1 se va da comanda: `docker swarm init`
 - Copiați comanda care vă apare după rularea docker swarm init pe celelalte două servere.
 - **Verificare:** Pe server1 rulați `docker node ls`. Ar trebui să vedeți toate 3 mașinile.
8. Creați o rețea de tip overlay care va rula peste swarm, aceasta va oferi funcționalitatea de DNS dar și load balancing:
- Rulați `docker network create -d overlay my-swarm-network`
 - **Verificare:** `docker network ls`
9. Creați un registry și puneți imaginea creată din Dockerfile acolo:
- Celelalte servere nu au acces la imaginile primului. Astfel trebuie să creăm un spațiu unde să punem imaginea care să fie accesibil de peste tot, acesta este un registry (e asemănător cu ce oferă în spate dockerhub).
 - `docker service create --network my-swarm-network --name registry --publish published=5000,target=5000 registry:2`
 - **Verificare:**
 1. `docker container ls` pe toate 3 serverele. Unul din ele va avea containerul registry.
 2. `wget http://127.0.0.1:5000/v2/_catalog/` urmat de `cat index.html`
Registry comunică peste http deci putem să îl interogăm așa. Ar trebui să vedem imaginea adăugată de noi.
10. Porniți un container redis și 3 containere de server web, ca servicii peste swarm:
- Porniți serviciu pentru redis:
`docker service create --name myredis --network my-swarm-network redis`
 - Adăugăm imaginea la registry. Din directorul în care se află fișierele din laborator (Dockerfile și index.php):
`docker build -t IMAGE_NAME` . E important să **nu uitați de punct**.



```
docker tag IMAGE_NAME 127.0.0.1:5000/IMAGE_NAME
docker push 127.0.0.1:5000/IMAGE_NAME
```

- Porniți serviciu pentru serverul web cu 3 containerele, una pe server:

```
docker service create --name SERVICE_NAME --network my-swarm-network -p 80:80 --replicas 3 --replicas-max-per-node 1 127.0.0.1:5000/IMAGE_NAME
```

- **Verificare:**

1. Pe server1: `docker service ls`
2. Pe toate serverele: `docker container ls`
3. Din browser intrați pe IP_SERVER_2 și IP_SERVER_3.
4. Se vor completa câmpurile și se va da refresh cu SHIFT apăsat.
5. Ar trebui să vedeți:

Hello World
Shared Value:

Private Value:

11. Modificați containerul anterior (Dockerfile) astfel încât să nu mai primiți eroare de fișier lipsă.

- Adăugați noua imagine în registru, sub un alt nume.
- Rulați un nou set de 2 containere folosind portul 88.

Exercițiile de la 1 la 11 sunt obligatorii. Conceptele explorate sunt esențiale pentru obținerea notei **minime** de promovare.

Vă recomandăm, pentru a crește șansele de a obține o notă cât mai mare să explorați și următoarele exerciții:

12. Scrieți fișierul Dockerfile și construiți containerul care să conțină proiectul vostru de la tehnologii web.

- Containerul ar trebui să aibă minim un server apache, dar poate fi necesar și php/mysql, în funcție de stadiul proiectului.
- Portul 80 va fi deschis pe mașinile fizice pe portul 8080, pentru a permite funcționarea simultană a acestor servere cu cele de la exercițiile anterioare.

13. Porniți acest container peste clusterul Docker Swarm folosind 4 replici ale sale.

Laboratorul va fi prezentat. Veți intra pe rând pe teams. Va trebui să aveți terminal putty deschis la mașina principală și să dați `docker service ls`. De asemenea va trebui să aveți deschis site-ul Azure și site-ul 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.