(https://profile.intra.42.fr/searches)

SCALE FOR PROJECT DOCKER-1 (/PROJECTS/DOCKER-1)

You should evaluate 1 student in this team

★

Git repository

vogsphere@vgs.42.us.or 

Introduction

We would love to follow the guidelines for
a smooth evaluation: Please remain courteous, polite, respectful and constructive
during this exchange. community and
the goals of 42 depends on it. \ r \ n Discuss and Clearly explain issues and defects
to the person you are Evaluating. \ r \ n there Will Sometimes differences be based
on the interpretation of the subject's \ r \ n requests or the scope of the features.
Stay open-minded to the vision of the younger and grade him as honestly as
possible. Wishing you a good defense!

Guidelines

REMEMBER YOU NEED TO EVALUATE ONLY THE CODE IN THE STUDENT'S REPO. \ R \ nMake
sure to \ "git clone \" the repository, and evaluate what is in there. \ R \ n \ nIf the
corrector has not yet done it, it is obligatory to read the entire subject before
beginning this defense.

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/1008/docker.en.pdf)

Topic (https://cdn.intra.42.fr/pdf/pdf/1007/docker.fr.pdf)

preliminaries

Preliminary instructions

Before you begin, take care to check the following:
- There is a rendering (in the deposit git), respecting the hierarchy requested
- No cheating, the student must be able to explain typed commands.

If an item in this list is not respected, the notation
stops there. Use the appropriate flag. You are encouraged to
continue discussing the project, but the scale is not
applied.

☑ Yes ☒ No.

Setting up the work environment

Check that the following points are correctly applied (do the necessary if necessary)

- The "docker version" command displays the version of docker and docker-machine
- Virtualbox is installed on the dump correction
- A symbolic link allows to re-read the .docker and VirtualBox VMs folders from the goinfre
to the home.

☑ Yes ☒ No.

How to Docker

This part allows you to evaluate the first part of the subject. For optimization questions, you will note the rendering by set of questions answering a thematic. For each command, you will have to execute it via your shell by doing `cat 01` or $ (cat 01). To go on good bases, you will already remove the virtual machine Char if it exists.

Before you start

Check that:
- 01: A virtual machine has been created in the name of Char with
the driver virtualbox (docker-machine ls)

- 02: You should have displayed something like 192.168.99.xxx

- 03: the command " env "in your shell displays 4 variables" DOCKER_ * ".
On the other hand, no command of assignment of variable of environment must
be present in the script, otherwise it is zero and end of the correction.

If one of these points is not respected, this part is counted false and you move on to the next

☑ Yes ☒ No.

My first container (04-09)

Check that:
- 04: hello-world image is available by docker images (without errors)

- 05: A greeting is displayed on the terminal.

- 06: Once the container is launched, check that:
* The container is named well overlord (docker ps)
* The port 80 of the container is well binded to the port 5000 of the virtual machine (docker ps)
* The command "curl http : //: 5000 "returns HTML corresponding to the nginx test page
* The command" docker inspect -f "{{.HostConfig.RestartPolicy}}" overlord "returns well {always ..}

- 07: An IP address of the form "172.X.0.X" appears (verifiable with an inspected docker)

- 08: Once the alpine container is launched, check that:
* You have the prompt "/ \ #"
* A whoami on this console you gives a good "root"
* An "exit" leaves the container well
* A docker ps -a does not find any trace of an alpine container

- 09: Once the debian container is launched, check that:
* You have the prompt "root @ hostname: / \ #"
* The execution of the commands given in the file is executed.
* You have the possibility to make a "git clone https://github.com/docker/docker.git"
* A small program in C should be compilable and executable in the context of the container (install vim if necessary)
* A docker ps -a can not find any trace of a debian container.

If one of these points is not respected, this part is counted false and you move on to the next

☑ Yes ☒ No.

Wordpress (10-18)

Check that:
- 10-11: the volume "hatchery" is well created, and that the command of 11 shows you a volume "hatchery" in addition to the others

- 12-13: Once the container mysql launched:
* You go back to your classic shell (only the digest of the launched container appears)
* The command 13 shows that MYSQL_ROOT_PASSWORD is set to "Kerrig @ n" and that MYSQL_DATABASE is well set to "zerglings"
* A "inspect spawning-pool docker" shows you that a "hatchery" volume is mounted on the destination "/ var / lib / mysql" (Key "Mounts")
* The command "docker inspect -f" {{.HostConfig.RestartPolicy}} "spawning-pool "returns well {on-failure .. }
* The "docker exec -it spawning-pool mysql -uroot -p" command requires you to enter a password
* The password is "Kerrigan" and the mysql command prompt is well visible

* the SQL command "show databases" shows a database "zerglings"

- 14: Once the wordpress container launched:
* You return well on your classic shell (only the digest of the launched container appears)
* You have possibility to launch a browser and go to http : //: 8080, and configure Wordpress so that the db used is that of spawning-pool
* Try a connection on
- 15: Once the container phpmyadmin launched:
* You return well on your classic shell (only the digest of the launched container appears)
* You have possibility to launch a browser and go to http: //: 8081
* You can access the available database on spawning pool and check that you have created wordpress tables

- 16: The mysql container logs are live

- 17: It's all about it

- 18: Using the command "docker exec -it overlord / bin / sh -c" kill 1 "", then use the command

"docker inspect -f '{{.RestartCount}}' overlord" which should then have incremented by 1.

If l one of these points is not respected, this part is counted false and you go to the next

Yes No.

Abathur (19)

Complete all the orders of the exercise to set up the container.
The container must embed the Flask framework (pip install Flask).
A python script must be present in the shared folder between the host and the container.
Launch this script from the context of the container,
so go to http: //: 3000 will post Hello World in title.
Likewise, logs must appear on the terminal.

If one of these points is not respected, this part is counted false and you move on to the next

Yes No.

The Swarm (20-30)

Check that:
- 20: a "docker node ls" shows Char in the HOSTNAME and his MANAGER STATUS is well Leader

- 21: same punishment with the question 01.

- 22: a "docker node ls" Aiur well displays in HOSTNAME and its MANAGER STATUS is other than Leader

- 23: command 23 works.

- 24-25: once the service rabbitmq launched:
* the `25` command displays the service" orbital-command "in 1 replica on 1 in replicate mode on a rabbitmq image: latest
* a" docker service ps orbital-command "shows you the status service" Running "
* a" docker service inspect -f "{{.Spec.TaskTemplate.ContainerSpec}}" orbital-command "
of environment which set a user and a specific password

- 26-27: once the engineering-bay service launched:
* a "docker service ps engineering-bay" shows you well the services in status "Running" with 2 replicas of made
* a "docker service inspect -f" {{.Spec.TaskTemplate.ContainerSpec}} "engineering-bay" gives you two environment variables that set a user and a password that allow connection to the orbital-command
* service command `27` makes you scroll through the logs of one of the 2 tasks of the service ... and shows you many zerg attacking orbital-command

- 28-29: once the marine service launched:
* a "docker service ps marines" shows you the services in status "Running" with 2 replicas of made
* a "docker service inspect -f" {{.Spec.TaskTemplate.ContainerSpec}} "marine" gives you two variables
of environment which set a user and a password which allow the connection to the orbital-command service
* to scrutinize the logs of the marine service shows well that the marines are in the process of collapse of the Zerg

- 30: a "docker service ps marines" displays you well the services in status "Running" with 20 replicas of made.
The service itself is not stopped, but well and truly updated.

If one of these points is not respected,

☑ Yes ☒ No.

Viscera Cleanup Detail (31-34)

All orders in this part must only make one line. (go there with great reinforcement of wc -l)

Check that:
- 31: a "docker service ls" no longer displays any service regardless of the state of it

- 32: a "docker ps -a" n displays no more containers regardless of the state of it

- 33: a "ls image docker" no longer displays any image

- 34: a "docker-machine ls" no longer displays the Aiur virtual machine

If the one of these points is not respected, this part is counted false and you go to the next

☑ Yes ☒ No.

DockerFiles

This part allows you to evaluate the second part of the subject. You will have to build each Dockerfile and evaluate the good implementation of the application To start on good bases, you will already start on a good basis. Either you redo a virtual machine, or you use Docker for Mac, either. The important thing is that a `docker ps -a` in the terminal shows you absolutely nothing.

Vim // Emacs

Build this dockerfile and launch it.
Vim or emacs must be launched, and the "explorer" mode of the editor must show you that you are in the context of the container and not your host.
Do the necessary tests.

If one of these points is not respected, this part is counted false and you move on to the next

☑ Yes ☒ No.

BYOTSS

Build this dockerfile and launch it.
It must appear in the background.
You can easily connect with a classic TeamSpeak client on it (take it on the MSC if it is not installed)

If one of these points is not respected, this part is counted false and you move on to the next one

☑ Yes ☒ No.

Dockerfile in a Dockerfile ... in a Dockerfile?

Build the dockerfile and pushez somewhere (hub docker, local registry ...).
Take advantage of this moment to create a blank Rails application in the directory (take a Ruby container and do the necessary).
Copy the dockerfile of the subject, try a build and launch the container with the necessary (expose of port, mode detache ...).
Check that you can access the Rails application by trying to access the IP of the machine via the exposed port.

If one of these points is not respected, this part is counted false and you move on to the next

☑ Yes ☒ No.

Salad Tomatoes Onions

Build the dockerfile and launch it with the necessary (port display, detach mode ...).
Make sure that Gitlab is available, that you can create some rest on it and that you can push as much in HTTPS as in SSH.

If one of these points is not respected, this part is counted false and you move on to the next

☑ Yes ☒ No.

---

bonus

Bonuses should only be assessed if and only if the mandatory game is PERFECT. By PERFECT, we obviously hear that it is fully realized, it is not possible to put his behavior in default, even in case of error, as vicious as it is, misuse, etc. Concretely, this means that if the mandatory part did not obtain ALL the points during this defense, the bonuses must be completely IGNORED.

🎵 I feel it coming ... I feel it coming ... I feel it coming ... I feel it coming ... 🎵

It's up to you to evaluate the different dockerfiles of the `02_bonus` folder.

The allocation of points is at the free discretion of the corrector.

Rate it from 0 (failed) through 5 (excellent)

5

---

ratings

Do not forget to check the flag

✔ Okay ★ Outstanding project

Empty work Incomplete work cheat Crash Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

General term of use of the website (https://signin.intra.42.fr/legal/terms/6)
Privacy policy (https://signin.intra.42.fr/legal/terms/5)
Legal notices (https://signin.intra.42.fr/legal/terms/3)
Declaration on the use of cookies (https://signin.intra.42.fr/legal/terms/2)
Terms of use for video monitoring (https://signin.intra.42.fr/legal/terms/1)
Rules of procedure (https://signin.intra.42.fr/legal/terms/4)