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

Comments

Introduction

We would like you to adhere to the following guidelines for a smooth evaluation:

- Please remain courteous, polite, respectful and constructive during this exchange. The relationship and trust between the community and the goals of 42 depend on it.

- Discuss and clearly explain issues and defects to the person you are evaluating.

- There will sometimes be differences based on the interpretation of the subject's requests or the scope of the features. Stay open-minded to the vision of the cadet \r\n and grade him as honestly as possible.

Wishing you a good defense!

Disclaimer

Guidelines

REMEMBER YOU NEED TO EVALUATE ONLY THE CODE IN THE STUDENT'S REPO.

Make sure to \"git clone\" the repository, and evaluate what is in there.

If 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/5023/docker.en.pdf)

Preliminaries

Preliminary instructions

Before you begin, be sure to check the following:
- Obviously, there is a repo (in the cloned git), respecting the subject
- No cheating, the student must be able to explain all the commands it turned in.

If an item in this list is not met, this evaluation stops there.
Use the appropriate flag. You are encouraged to debate on the rest of this evaluation, but no more marks should be applied."

☑ Yes   ✕ No

Setting up the working environment

Verify that the following instructions work correctly (run the necessary commands)

- The \"docker version\" command displays the docker and docker-machine version
- Virtualbox is installed on the machine
- Symbolic links are set for the .docker and the VirtualBox VMs image directories to somewhere you have space.

☑ Yes   ✕ No

How to Docker

Before you begin

This part is about evaluating the first part of the subject.
For optimization questions, you will evaluate the output by running the

commands in the subject.

For each command, you will have to execute it via your shell by running
`cat 01` or $(cat 01).

Make sure you start from scratch by removing the Char
virtual machine if it already exists.

⊘ Yes   ✕ No

docker-machine (01-03)

Make sure that:

- 01: A virtual machine Char has been created with the virtualbox
driver (docker-machine ls)

- 02: You should see something like 192.168.99.xxx

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

If any of these points is false, this part is counted as false"

⊘ Yes   ✕ No

My first container (04-09)

Make sure that:

- 04: The ``hello-world'' image is available when you enter
`docker images` (without errors)

- 05: A greeting message is displayed on the terminal.

- 06: Once the container is launched, check that:
* The name of this container is overlord (docker ps)
* Port 80 of the container is well binded to port 5000 of the
virtual machine (docker ps)
* The command \"curl http://:5000\" returns the test
page of nginx

* The \"docker inspect -f\" {{.HostConfig.RestartPolicy}}
\"overlord\" command returns {always ..}

- 07: An IP address of the form \"172.X.0.X\" appears
(that you can check with `docker inspect`)

- 08: Once the Alpine container is launched, check that:
* The prompt \"/ #\" is displayed
* A `whoami` command prints \"root\"
* An `exit` command stops the container and bring
you back to your host console
* A `docker ps -a` command returns no traces of this container

- 09: Once the debian container is launched, check that:
* You are prompted \"root @ hostname: / #\"
* The execution of the commands given inthe file executes
* You can run the command \"git clone
https://github.com/docker/docker.git\" without errors
* A small program in C should be compileable and runnable
in the context of the container (install vim if necessary)
* A `docker ps -a` command returns no traces of this container

If any of these points are not met, this part is counted
wrong and you move to the next one

⌒ Yes   ✕ No

Wordpress (10-18)

Make sure that:

- 10-11: The volume \"hatchery\" is created, and that the command #11
shows you a \"hatchery\" volume in addition to the others

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

* The password is \"Kerrigan\" and the command prompt mysql is visible
* The SQL command \"show databases\" shows a database \"zerglings\"

- 14: Once the wordpress container is launched:
  * You come back to your classic shell (only the digest of the container launched appears)
  * You can launch a browser and go to http://:8080, and configure Wordpress so that the db used is that of spawning-pool
  * Install Wordpress and verify that the app is properly installed

- 15: Once the container phpmyadmin launched:
  * You come back on your classic shell (only the digest of the container launched appears)
  * You can launch a browser and go to http://:8081
  * You can access the available database on spawning-pool and verify that you have created wordpress tables

- 16: Logs of the mysql container are displayed live

- 17: It talks about him

- 18: Use the command `docker exec -it overlord /bin/sh -c \"kill 1\"`, and then `docker inspect -f '{{.RestartCount}}' overlord` should be incremented of 1.

If any of these points are not met, this part is counted wrong and you move to the next one

⊘ Yes   ✕ No

Abathur (19)

Execute all the exercise commands in order 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 accessing the URL http://:3000 will display Hello World title.
Similarly, logs must appear on the terminal.

If any of these points are not met, this part is counted wrong and you move to the next one

⊘ Yes   ✕ No

The Swarm (20-30)

Make sure that:

- 20: a \"docker node ls\" displays Char in the HOSTNAME and its
MANAGER STATUS is Leader

- 21: Same answer as with question 01.

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

- 23: the command 23 operates.

- 24-25: Once the RabbitMQ service is launched:
* Command 25 displays the \"orbital-command\" service on 1/1 replica in
replicate mode on a rabbitmq: latest image
* A \"docker service ps orbital-command\" shows you the service in
status \"Running\"
* `docker service inspect -f \"{{.Spec.TaskTemplate.ContainerSpec.Env}}\"
orbital-command` gives you two environment variables that set a user
and a specific password

- 26-27: Once the engineering-bay service has been launched:
* A `docker service ps engineering-bay` displays well the services
in status \"Running\" with 2 replicas of made
* `Docker service inspect -f \"{{.Spec.TaskTemplate.ContainerSpec.Env}}\"
engineering-bay` gives you two environment variables that set a user
and a password to connect to the orbital-command service
* The command #27 makes you scroll the logs of the 2 tasks of the
service ... and shows you many zergs attacking orbital-command !!!

- 28-29: Once the marine service launched:
* A `docker service ps marines` displays the services in status
\"Running\" with 2 replicas
* A `docker service inspect -f \" {{.Spec.TaskTemplate.ContainerSpec.Env}}\"
marines` gives you two environment variables that set a user
and a password that allow connection to the orbital-command service
* A `docker service logs marines`shows that marines are pwning some Zergs\

- 30: A `docker service ps marines` shows the services
in status \"Running\" with 20 replicas. The service itself
is not stopped, just updated.

If any of these points are not met, this part is counted
wrong and you move to the next one

⊘ Yes   ✕ No

Viscera Cleanup Detail (31-34)

All commands in this part must each be one line.
(Check with the mighty wc -l)

Make sure that:

- 31: a `docker service ls` shows no service
- 32: a `docker ps -a` shows no container
- 33: a `docker images ls` shows no image
- 34: `docker-machine ls` shows that Aiur is definitely destroyed

If any of these points are not met, this part is counted
wrong and you move to the next one

⊘ 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 correct
implementation of the application. You need to start fresh. Either you make a virtual machine, or you use Docker for Mac, your
choice. The important thing is that a `docker ps -a' in the terminal shows you absolutely nothing.*

Vim // Emacs

Build this dockerfile then run it.
Vim or Emacs must start on its own, and the editor's \"explorer\"
mode must show you that you are in the context of the container
and not your host.
Make the necessary tests.
If any of these points are not met, this part is counted wrong
and you move to the next one.

⊘ Yes   ✕ No

BYOTSS

Build this dockerfile then run it.
It must appear in the background.
Check that this image is not the offical docker's one, if not -42.
You can easily connect with a classic TeamSpeak client on it
(take it on the MSC if it's not installed)

If any of these points are not met, this part is counted
wrong and you move to the next one

☑ Yes   ✕ No

Dockerfile in a Dockerfile... in a Dockerfile ?

Build the dockerfile and push it somewhere (Docker Hub, local registry ...).
Take advantage of this moment to create a blank Rails application in
the directory (take a container and do the necessary).
Copy the dockerfile of the subject, try to build and launch
the container with the necessary (exposed ports, detach mode ...).
Verify that you can access the Rails application by try accessing
the machine's IP through the exposed port.

If any of these points are not met, this part is counted
wrong and you move to the next one"

☑ Yes   ✕ No

What does the fox say ?

Build the dockerfile and launch it with the necessary
(exposed ports, detached mode ...).
Make sure that Gitlab is available, that you can create users
and repo on it and that you can push as much in HTTPS and SSH.

If any of these points are not met, this part is counted
wrong and you move to the next one"

☑ Yes   ✕ No

BONUS

Bonuses must be assessed if and only if the Mandatory part is PERFECT. By PERFECT, we mean the mandatory is fully realized and it is not possible to find an error. Concretely, this means that if the mandatory part did not get ALL 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 is up to you to evaluate the different dockerfiles of the
`02_bonus` folder.

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

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

Ratings

Don't forget to check the flag corresponding to the defense

✔ Ok   ★ Outstanding project

📄 Empty work   📄 Incomplete work   📑 Cheat   💥 Crash   🚫 Forbidden function

Conclusion

Leave a comment on this evaluation

Preview!!!

General term of use of the site
(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 surveillance
(https://signin.intra.42.fr/legal/terms/1)

(https://sign