TSR / NIST

3rd Lab Project

This exam consists of 10 multiple choice questions. In every case only one answer is correct. You should answer in a separate sheet. If correctly answered, they contribute 1 point to the exam grade. If incorrectly answered, the contribution is negative: -0.333. So, think carefully your answers.

1. The last task proposed <u>in the first part</u> of this lab project consists in running this command in a basic version of the "cbw" application:

```
docker-compose up --scale cli=8 --scale wor=3
```

What is the result of that command?

a	Some errors arise and no container is started, since the Docker images to be used in that command are "client" and "worker", instead of "cli" and "wor". False. Although the images of those components have the names mentioned in that statement, the "docker-compose.yml" file gives those elements the names "cli" and "wor", respectively. Therefore, no error is generated by that command, since it has been correctly written.
b	Some containers are started but clients are unable to interact with workers, since no broker has been started. False. No broker is mentioned in that command, but the broker component is already mentioned and managed in the "docker-compose.yml" file used in that part of this lab project. With that, when the "docker-compose up" command is run, one broker is started by default. Such broker makes possible the communication between clients and workers.
[✓ c]	A broker manages the interactions among eight clients and three workers. True. That is the actual result of that command.
d	A broker, a client and a worker are started. Depending on the host workload, the Docker daemon may start up to 7 additional clients and up to 2 additional workers. False. That command starts eight clients, one broker and three workers. The result of that action does not depend on the host workload.

2. <u>In the second part</u> of this lab project a "logger" component was added to the "cbw" application. Its inclusion demanded this new block in the "docker-compose.yml" file:

```
log:
  image: logger
  build: ./logger/
  volumes:
    - /tmp/logger.log:/tmp/myfile.log
```

Which is the correct final content in the broker-related part of that "docker-compose.yml" file?

a	None, because there is no broker-related section in that file. False. There is a broker section in the file.

b	`bro:`
	`  image: broker`
	`  build: ./broker/`
	False. That was the initial contents of that section in the versions where not "logger" component was used. However, when a "logger" is run, each of the other components should specify a "link" with it in order to build its appropriate URL. That URL is used as an argument for starting those other components.

c	`bro:`
	`  image: broker`
	`  build: ./broker/`
	`  links:`
	`    - wor`
	False. Although the broker interacts with workers, it does not need any "link" to them since the broker binds the ports used in that kind of communication. Because of this, the workers are the components that must specify a link to the broker, instead.

[✓ **d**]	`bro:`
	`  image: broker`
	`  build: ./broker/`
	`  links:`
	`    - log`
	`  environment:`
	`    - LOGGER_URL=tcp://log:8066`
	True. Since the logger binds its PULL socket that behaves as its endpoint to be used by the remaining components, those other components have a dependence on it. Such dependence is resolved using the "links:" clause, as shown in this part. Besides, taking that link as a base, the value for the LOGGER_URL environment variable is also built in the last block of this fragment.

3. Considering the "logger" block from the "docker-compose.yml" file shown in the previous question, why is the "volumes" section needed there?

[✓ **a**]	Because the "/tmp/myfile.log" file in the container of that component is mapped to the "/tmp/logger.log" file in the host. True. The goal of a "volumes:" section is to set the correspondence between directories or files in the host (that are specified first) and directories or files in the container (that are specified after the colon).
b	Because the "docker-compose.yml" file is now placed in the "/tmp" folder and its name may be "logger.log" or "myfile.log". False. The "volumes:" clause is not related to the location of "docker-compose.yml" files or their names.
c	Because that component accesses the CD-ROM while it is running, and the contents of the CD to be read are taken from a file called "/tmp/logger.log" or "/tmp/myfile.log". False. The "volumes:" clause is not related to files that contain images of storage devices.
d	Because the "logger" Dockerfile is now placed in the "/tmp" folder and its name may be "logger.log" or "myfile.log". False. The "volumes:" clause isn't related to the location of Dockerfiles or their name.

4. Which of the following statements about the "logger" component to be deployed in the <u>second part</u> of the lab project IS FALSE?

a	It receives the trace messages generated by the other components.

	True. That is the functionality of the "logger" component.
b	It handles its communication with other components using a PULL ØMQ socket. True. It binds its PULL ØMQ socket and other components should send their messages to it, using a PUSH socket connected to that endpoint.
[✓ **c**]	It handles its communication with other components using a file placed in a volume. False. Communication with other components is managed using ØMQ sockets.
d	It handles its communication with the user using a file placed in a volume. True. All trace messages are written to a "log" file. That log file may be read by the user accessing to a "volume" in the host.

5. In that <u>second part</u> of the lab project, let us assume that we have started 2 clients of type B, 1 worker of type B, 1 broker but 2 loggers instead of a single one. In that case, which of the following statements correctly describes the behaviour of that deployment?

a	The second logger is unable to start, since it cannot bind its ØMQ socket (the corresponding port number is already busy). An error is reported by docker-compose. False. Each instance of a Docker container/component has its own set of hardware resources. The container ports used by a given instance do not affect the set of free container ports in other instances. Therefore, the port mentioned in this part of the question cannot be busy in the second logger.
[✓ **b**]	Clients, broker and worker connect to the first logger and nothing strange happens in that execution. The second logger is ignored. True. That is the behaviour shown in that scenario.
c	Each logger creates a different log file. Some components connect with the first logger and some others with the second one. Messages may be found in both files. False. The "links:" clause in the "docker-compose.yml" file only manages connections with the first instance of the linked component. Other instances are not considered. Because of this, all components connect with only one of the loggers. Besides, only the log file of that logger is mapped to the host. The other file remains empty.
d	The worker is unable to manage so many clients and it crashes once started. No worker-related message may be found in the log. False. A worker may serve as many clients as needed. There is no limit on this. It will not crash due to an excessive workload level.

6. In that <u>second part</u> of the lab project, let us assume that we have started 2 clients of type B, 2 workers of type B, 1 logger but 2 brokers instead of a single one. In that case, which of the following statements correctly describes the behaviour of that deployment?

a	The 2nd broker is unable to start. It cannot bind its ØMQ sockets (the corresponding port numbers are already busy). An error is reported by docker-compose. False. Each instance of a Docker container/component has its own set of hardware resources. The container ports used by a given instance do not affect the set of free container ports in other instances. Therefore, the ports mentioned in this part of the question cannot be busy in the second broker.

b	The logger is unable to manage so many components and it crashes once started. No trace-related message may be found in the log. False. A logger may serve as many reporting components as needed. There is no limit on this. It will not crash due to an excessive workload level.
c	Some clients and workers connect with the first broker and some others with the second one. The workload is shared by both brokers. False. The "links:" clause in the "docker-compose.yml" file only manages connections with the first instance of the linked component. Other instances are not considered. Because of this, all components connect with only one of the brokers. The other broker is not used.
d	Clients and workers connect to the first broker and nothing strange happens in that execution. The second broker is ignored. True. That is the behaviour shown in that scenario.

7. In the <u>third part</u> of this lab project, its first task demands that clients do not run in the same host where broker and workers run. In that scenario, which is the FALSE statement?

a	No client-related section exists in the docker-compose.yml file. True. In that scenario, client components cannot be managed by the "docker-compose.yml" file, since the docker-compose command may only control a set of components placed in the same host. Therefore, no client-related section may be found in that file.
b	A "ports:" statement must be in the broker section of the docker-compose.yml file to assign a host port number to the frontend port of the broker container. True. Clients may only connect to the broker through a public port number in the host where all other components (broker and workers) run. To this end, a "ports:" instruction is needed in the file section of the broker.
c	Clients may be started from the command-line interface in another computer, passing them an appropriate broker URL. True. That is the goal of that task.
d	The broker-related URL to be used by clients uses as its IP address that of the broker container. False. When clients are placed in another computer, they have no access to the Docker network that interconnects the containers that run in another host. Because of this, the IP address of that broker container cannot be accessed from that other computer. Those clients need to know the IP address of the host where the broker container runs and the frontend endpoint for that broker must be mapped to one of the ports of the host. Let us refer to that host port as P1. The URL to be used consists of those two parts: host-IP & host-port-P1.

8. In the <u>third part</u> of this lab project, a "worcli" component is presented and used. The goal of that component is to show...:

a	...that the same component may appear in two different docker-compose.yml files and that both files may be used for scaling it while it is running. False. The docker-compose command cannot handle a scenario like that.

b	...that a component cannot behave simultaneously as a client for a service and a worker for another. False. A component may have such behaviour. Indeed, that is the behaviour being shown by “worcli”.
c	...the difficulties that arise when a scalable component behaves as a client for a service and a worker for another. True. In that case, its client dependences cannot be administered or resolved using a docker-compose.yml configuration file.
d	...that a given component may scale while it behaves as a worker for a service but not when it behaves as a client for another service. False. “Worcli” may scale and it plays both roles.

9. The 2nd lab project presented the heart-beating approach for managing worker failures. Let us assume that we can chain services: A’s workers are clients of service B. Which of these aspects could be critical in order to set the failure detection interval managed by A’s broker?

a	The longest service time in service B, since it conditions the resulting service time in its callers. True. Those callers are the workers for service A. If brokers are using heart-beating in order to identify when one of their workers have failed, the deadline for such A’s workers depends on their own service time and on the service time of any other services invoked by them. In this case, these other invoked services are the operations served by B’s workers.
b	The amount of job classes. False. In these examples all job classes have the same service time. Only the variations in service time may have any effect on the heart-beat deadlines.
c	Dockerfile for A’s broker. False. In the second and third lab projects there was no configuration parameter or environment variable related with the heart-beating management in the broker’s Dockerfile.
d	The amount of workers in each service, since they condition the response time of each forwarded request. False. The worker service time is directly specified through an argument when the worker is started. Thus, the amount of workers being used has not influence on the service time for each request being received by a worker.

10. Considering the chained service from the previous question, where broker and worker components cannot be executed at desktop nodes, its worcli component may be executed:

a	In any node (desktop, host for A deployment, host for B deployment). False. The “worcli” component behaves as a worker for service A and as a client for service B. In order to be easily managed (and scaled, when needed), the instances of “worcli” should appear in the “docker-compose.yml” of service A. Therefore, “worcli” instances must be placed in the host where service A has been deployed.

b	In the host for A deployment. True. See the explanation in the first choice of this question.
c	In the host for B deployment. False. See the explanation in the first choice of this question.
d	In the desktop. False. See the explanation in the first choice of this question.