

Speakers:

Mário Dagot

Jorge Dias

Hugo Calado



From Zero to Docker

Hands On with Docker

#	Summay/Command	Purpose
1	docker install	Quick overview on how to install docker on Linux (apt-get) and Windows (chocolatery) Instructions on: https://docs.docker.com/install/linux/docker-ce/ubuntu/
2	docker images	List images. Show that there's nothing currently.
3	docker hub	Show docker hub and talk a bit about the public and private registries. Show some images on docker hub: tomcat, mysql, postgres, wordpress Explore the image page: Show command pull, versions available, documentation, Dockerfile
4	docker pull tomcat:latest Alternativa: cd /opt/fromzerotodocker/backup_images docker load -i ./tomcat_latest.tar	Pull tomcat image
5	docker images	List images. See that now we have an image locally
6	docker container ls	List containers. Show that there's nothing there. Talk about image vs container.
7	docker run -d tomcat docker ps	List containers. Show the new container running. Talk about "funny" name given. Talk about multiple running containers (instances) based on the same image
8	docker logs <funny name>	See that tomcat started with success and that the logs reflect that.

9 Check tomcat home page	Access tomcat home page on the browser: http://localhost:8080. Show that we cannot access the running instance. Talk about how containers cannot be accessed from the host machine without exposing their internal ports to the outside world.
10 docker rm <funny name>	Show how to remove a container. We should get an error because the container is running. Option 1: docker stop <funny name>; docker rm <funny name> Option 2: docker rm -f <funny name>
11 docker run -p 8080:8080 tomcat	Run the container now exposing internal port 8080 to host port 8080. Access the tomcat home page. It should work now.
12 docker ps	List containers and show that the port exposed is not visible
13 docker run -d --name tomcat8081 -p 8081:8080 tomcat docker ps	Run another instance of tomcat and give it a custom name. Show two instances running. Access tomcat home page.
14 docker stop tomcat8081	Stop tomcat8081 container.
15 docker ps -a	List containers. See that tomcat8081 is gone. Explain that stopped containers are not listed by default. List all containers.
16 docker start tomcat8081	Start container
17 docker exec -it tomcat8081 bash mkdir /usr/local/tomcat/webapps/test echo "Hello World." > /usr/local/tomcat/webapps/test/index.html	Access inside the container. Navigate the filesystem. Show tomcat installation dir: /usr/local/tomcat Change something inside tomcat8081. Create test application and access.
19 docker stop tomcat8081 docker rm tomcat8081 docker run -d --name tomcat8081 -p 8080:8080 tomcat docker exec -it tomcat8081 bash	Show that changes were lost. Containers are state persistent, but images aren't. A container is always created from a image and changes done on container are lost if the container is removed.
20	Discuss about volumes. Give the example of a Database. It's necessary to guarantee the persistence of the data even if the container is removed.
21 docker volume create tomcat-logs docker volume ls	Create a volume and list it.

<pre> docker stop tomcat8081 docker rm tomcat8081 docker run -d --name tomcat8081 -p 8080:8080 -v </pre>		
22	tomcat-logs:/usr/local/tomcat/logs tomcat	Create a tomcat with logs shared on a volume.
	docker inspect tomcat8081	Inspect the new tomcat8081 container and show the volume.
23	docker inspect tomcat-logs	Inspect the volume tomcat-logs and show that tomcat logs are accessible from host.
24	docker logs -f tomcat-logs	Access to Web interface and check logs on docker volume.
<pre> sudo ls -l /var/lib/docker/volumes/tomcat-logs/_data sudo tail -f /var/lib/docker/volumes/tomcat- </pre>		
25	logs/_data/catalina.out	Show that is also possible to check logs on host machine accessing to folder identified on previous point.
<pre> docker rm tomcat8081 </pre>		
26	sudo ls -l /var/lib/docker/volumes/tomcat-logs/_data	Remove tomcat8081 container and check that tomcat logs are still there.
<pre> docker pull mysql/mysql-server Alternativa: cd /opt/fromzerotodocker/backup_images </pre>		
27	docker load -i ./mysql-mysql_server.tar	Now lets create database. First pull latest MySQL image
<pre> docker volume create mysql-data </pre>		
28	docker volume create mysql-initial-script	Create a volume to store MySQL data and initial Sql scripts
29	docker inspect mysql-data	Inspect mysql-data volume.
<pre> docker run --name=mysql1 -d mysql/mysql- server:latest -v mysql-data:/var/lib/mysql -v mysql- initial-script:/docker-entrypoint-initdb.d </pre>		
30	docker ps	Create container mysql1.
<pre> docker logs mysql1 </pre>		
31	docker logs -f mysql1	Check logs from created container.
<pre> docker logs mysql1 2>&1 grep GENERATED docker exec -it mysql1 bash mysql -uroot -p ALTER USER 'root'@'localhost' IDENTIFIED BY </pre>		
32	'password';	Access to newly created database and change root default password to 'password'. On image is point 26

cd /opt/fromzerotodocker	Go to docker compose folder and then to tomcat Dockerfile folder. Folder contains:
cd app	- app
ls -l	- war file
33 vi Dockerfile	- properties files
<pre> # Tomcat image version FROM tomcat:8-jre8 # User that maintains Dockerfile MAINTAINER dagotma # Add war file to webapps folder to auto deploy webapp ADD *.war /usr/local/tomcat/webapps/ # Copy properties files to a folder inside tomcat classpath ADD *.properties /usr/local/tomcat/lib/ # Tomcat command to execute when running Tomcat Container from this image </pre>	
34 CMD ["catalina.sh", "run"]	Add content to Dockerfile
35 docker build --tag tfc-tomcat .	Will create image tfc-tomcat .
36 docker images	Show new created tfc-tomcat image
cd ..	
37 vi docker-compose.yml	
38 version: 3.3	Define docker-compose api to be used.

39	<pre> services: app: build: ./app environment: - JAVA_OPTS=-Xms1024M -Xmx1024M ports: - 8080:8080 links: - db depends_on: - db volumes: - app_logs:/usr/local/tomcat/logs </pre>	Create app service, that will be created from na tomcat custom image, previously created.
40	<pre> db: image: mysql:5.7.23 restart: always volumes: - db_data:/var/lib/mysql environment: - MYSQL_ROOT_PASSWORD=password - MYSQL_DATABASE=gestaotfc - MYSQL_USER=gestaotfc - MYSQL_PASSWORD=tf ports: - 3306:3306 </pre>	Create db image from MySQL image.
41	<pre> volumes: app_logs: db_data: </pre>	Define volumes to be used on containers.
42	sudo docker-compose -d	Execute docker-compose.
43	docker ps	Validate new created containers.

Users:

p4997 - ROLE: Coordenador de TFCs

- p5617 - ROLE: Professor

- a111 - ROLE: Aluno

- a112 - ROLE: Aluno

44 Password: password