Command	Description	Key word
docker version	To show docker version	Version
docker info	To show all the containers running , paused and sttoped.	Info
docker login	To log to docker repository (Docker hub)	Login
docker ps	To show the conteiners that are running, you can run also docker ps -a   head	Show
	head = the last ten.	
	a = Also the containers that they are not running.	
docker run nameOfTheImage	Run the container, example: docker run alpine:3.7 ls -l and l can see all the files  gaston@gaston:**\$ sudo docker run alpine:3.7 ls -l Unable to find image 'alpine:3.7' locally 3.7: Pulling from library/alpine	Run
docker run -d nameOfTheImage	With -d I run the container and it continues running, we use -d when we run for example a nginx service.	Run
docker start idOfCointaner.	To start a container.	Start
docker stop idOfCointaner.	To stop a container.	Stop
docker rm idOfCointaner.	To remove a container.	Remove
docker kill idOfCointaner.	To kill a container	Kill
docker pull nameOfTheImage	To download a docker image	Pull / Download
docker run -it nameOfTheImage sh docker run -it nameOfTheImage /bin/bash	To run a shell in the container. i = interactive  gaston@gaston:~\$ sudo docker run -it alpine:3.7 sh / # 1s bin etc lib mnt root sbin sys usr dev home media proc run srv tmp var / # cd home /home # 1s /home # _	Run

docker exec -it IdOfTheContainer sh	To execute a container that it is running.	Execute		
docker exec -it IdOfTheContainer bin/bash				
docker commit IdOfTheContainer	To create a new image with our changes. Example: I ran an alpine container, I installed new packages and then I			
	close the container session, I can see the container that were running (docker ps -a) and create an image with the			
	command commit. Other example, You can install in ubuntu image figlet and then create your own image and			
	run it :			
	gaston@gaston:~\$ sudo docker run mydockerimage:1.0 figlet "My image has figlet"			
	gaston@gaston:~\$			
	In this example the image is taged with a name and version.	Show		
docker image Is	To show our images:			
docker images	gaston@gaston:~\$ sudo docker image ls REPOSITORY TAG IMAGE ID CREATED SIZE nello—world latest bf756fb1ae65 5 months ago 13.3kB alpine 3.7 6d1ef012b567 15 months ago 4.21MB gaston@gaston:~\$ _			
docker rmi idOfTheImage	To remove an image.	Remove		
docker image tag IdOfTheImage	To tag the image, you can run the command docker image tag IdOfTheImage myDockerImage:1.0 or docker			
	image tag IdOfTheImage myDockerImage , if you don't especify the version the default is latest.  gaston@gaston:~\$ sudo docker image tag d24ea63ab15e mydockerimage:1.0 gaston@gaston:~\$ sudo docker image ls REPOSITORY TAG IMAGE ID CREATED SIZE mydockerimage 1.0 d24ea63ab15e 2 minutes ago 97.3MB ubuntu latest 1d622ef86b13 7 weeks ago 73.9MB hello-world latest bf756fb1ae65 5 months ago 13.3kB alpine 3.7 6d1ef012b567 15 months ago 4.21MB gaston@gaston:~\$ _			

Dockerfile	Dockerfile is a set of instuctions to create images.			Dockerfile / Build	
	I create a file with inst	I create a file with instructions that update ubuntu and the install figlet:			
	gaston@gaston:~/d FROM ubuntu	gaston@gaston:~/docker\$ cat Dockerfile FROM ubuntu			
	RUN apt-get updat				
	Now I build a new ima				
	Sending build cont Step 1/2: FROM ub> 1d622ef86b13 Step 2/2: RUN apt> Using cache> a4aeecb77ad9 Successfully built Successfully tagge gaston@gaston:~/do REPOSITORY mydockerimage mydockerimage mydockerimage ubuntu hello-world alpine gaston@gaston:~/do	ext to Docker daemountu  -get update && apt-  a4aeecb77ad9 d mydockerimage:1.2 cker\$ sudo docker: TAG 1.1 1.2 1.0 latest latest 3.7 cker\$ sudo docker r	get install figlet –y	B B B	
docker image history IdOfTheImage	To check the history of	of the image:			History
	IMAGE	cker\$ sudo docker : CREATED	mage history a4aeecb77ad9 CREATED BY	SIZE	
	COMMENT a4aeecb77ad9	12 minutes ago	/bin/sh –c apt–get update && apt–get install…	23.4MB	
	1d622ef86b13	7 weeks ago	/bin/sh -c #(nop) CMD ["/bin/bash"]	ОВ	
	<missing></missing>	7 weeks ago	/bin/sh –c mkdir –p /run/systemd && echo 'do…	7B	
	<missing></missing>	7 weeks ago	/bin/sh -c set -xe && echo '#!/bin/sh' > /…	8118	
	<missing></missing>	7 weeks ago	/bin/sh -c [ -z "\$(apt-get indextargets)" ]	1.01MB	
	<missing></missing>	7 weeks ago	/bin/sh -c #(nop) ADD file:a58c8b447951f9e30	72.8MB	
docker logs IdOfTheContainer	To check the logs of the	he container. To add	the time you can run docker logs -t idOfTheContainer		Logs
docker stats IdOfTheContainer	To check the stats of t				Stats

docker system prune	To clean all what you are not using:	Clean / Remove
docker inspect IdOfTheContainer	gaston@gaston:~\$ sudo docker system prune WARNING! This will remove: - all stopped containers - all networks not used by at least one container - all dangling images - all dangling images - all dangling build cache  Are you sure you want to continue? [y/N] y Deleted Containers: 44e7a2d5c4d5s43bfefa2eb94e3d16797fa5039ddd30230386affe5974e5197b 97757b94ef680c7cc3985e5d9c94c5d3c91d47135f531af1879e19665b445d609 d66e0807244293cad8f2a25c49f141dfefg22ddes181faf262288fa56b3d9a6b 64e00228d91b3236284350bd52954a4c0ecec52a46e12070b0756e8cbe01164 laebc5f390c73f1918d246c613laebee31f7ebb1f270730ddd6a20a367e9d3ed5c 5a146203fa525124b176924af64637a493abde137cc24d2b2cd11705bac29975 lc25e5c5eb4859f7c505344031b133abbdd43b819a268lf3b8b868fb61bbab52 664adbdef87cfa6195027fc7dec4f2061037fd300eb5dec40a45b88aa110a 9f03fb060fbf439c2c7babf8d87d14a2f8313fd6134ed20d7ebfd10a2c40f1200 f35e977547a72ca7c3aeid2fdee7e26d8f1a70ccf39a333c73e95b6be0427d18 244dcd321170f5964323b494f81ebba5dd97ee8a4a7a4f922a3f5f34ad93d7bc c619teb18bb6aac8d887003355e3d70f160ad75db6445b0ab0bec97156df9b 045509ff725a72ac7aseid2fd46e7e26df1706ad75db6445b0ab0bec97156df9b 045509ff725a72ac7aseid2f2d646266203f30835c3e3b5bc097136fbbc09719bb 045509ff725e73d50b1be2195fbc30a988052859917365bcc091ee503930cb 6bb775d58c94643afd60236309933fbd0cfe0485800a73bf2e632e2b86f9199c 0246775a665995bc54562c85e1858496cded203f3d85ac430bcfc30d86ef1a23c6 d060cc7357ddf61f3c242ce9ba3da625875139266104315479d4dfdaff7702d 47e489533990c595994150c126f43db49f3756a55ca30bcfc30d86ef1a23c6 d060cc7357ddf61f3c242ce9ba3da625875139266104315479d4dfdaff7702d 47e489533990c59659185049464748756a562e362d2d295b00855793924b83735 0602b24ae985c5956593984150c126f43db49f3756a59caa47b6c03855793924b83735 0602b24ae985c996691869674644785e6e483abbc652662c2dd25b00856793924b83735 0602b24ae985c96691869634860f64044875ee6483abbc652662c2dd25b00856793924b83735 0602b24ae985c9669186963474756866674635a66a6483b0662796a62c2dd25b00856793924b83735 0602b24ae985c966598696986967423137da64fb1de880410a4d8230983eac Deleted Networks: dcker_default  Total	
docker inspect IdOfTheContainer	Give you information about the container (variables, ip , ports and more)	
docker cp	To copy files from the container to the host or from the host to the container.	Сору
	docker cp idOfTheContainer:/folderContainer/Subfolder/file.txt /home/gaston/folderhost/	
	docker cp /home/gaston/folderhost/file.txt idOfTheContainer:/folderContainer/Subfolder/	
docker add	To add files from a url to the container.	Add
	docker add mysite.com/file /files/	

Volume	When you run a container, you can create a volume to save files in your host server, for example:	Volume
	docker run -v home/gaston/docker/index.html:/usr/share/nginx/html/index.html:ro -d nginx:1.19.0 docker run -v path_inside_the_host:path_inside_the_container:xx -d web_server_image:version	
	-v = volume -d = run and continue running. ro = read only	
Ports	To open a port you can add in the docker run a new parameter:	Ports
	docker run -v home/gaston/docker/index.html:/usr/share/nginx/html/index.html:ro -p 8080:80 -d nginx:1.19.0	
	-p host_port:container_port	
	A now you can check from your host server your site in the container.	
	gaston@gaston:~/docker\$ sudo docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES	
	1aebc5f890e7 nginx:1.19.0 "/docker-entrypoint" 2 minutes ago Up 2 minutes 0.0.0.0:8080–>80/tcp practical_darwin gaston@gaston:~/docker\$ _	

Docker compose You can create a yaml script to work in a more organized way and then run the yaml to create the container: Docker compose gaston@gaston:~/docker\$ cat docker-compose.yaml version: '3.1' services: wordpress: image: wordpress:5.4.2-php7.2-apache ports: - 8080:80 environment: WORDPRESS\_DB\_HOST: mysql WORDPRESS\_DB\_USER: root WORDPRESS\_DB\_PASSWORD: root WORDPRESS\_DB\_NAME: wordpress links: - mysql:mysql #create a line in the host wordpress container to reference th ip of the mysql container, to avoid issues if the ip changes mysql: image: mysq1:8.0.20 command: --default-authentication-plugin=mysql\_native\_password environment: MYSQL\_DATABASE: wordpress MYSQL\_ROOT\_PASSWORD: root volumes: - /home/gaston/mysql-data:/var/lib/mysql gaston@gaston:~/docker\$ And then run your yaml script with the command docker-compose up -d gaston@gaston:~/docker\$ sudo docker–compose up –d Starting docker\_mysql\_1 ... done
Starting docker\_wordpress\_1 ... done
gaston@gaston:~/docker\$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STA NAMES PORTS d66e08072442 wordpress:5.4.2-php7.2-apache "docker-entrypoint.s.." 6 minutes ago

As you can see , you have 2 containers, one with mysgl and the other with wordpress.

0.0.0.0:8080->80/tcp docker\_wordpress\_1

docker\_mysql\_1

"docker–entrypoint.s…" 6 minutes ago

mysq1:8.0.20

3306/tcp, 33060/tcp

Docker push

13 seconds

18 seconds

4c00228d911

gaston@gaston:~/docker\$

To push a image to docker hub

Push