Command	Description	Key word
docker version	To show docker version	Version
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 I can see all the files gaston@gaston: **s 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 docker start idOfCointaner.	druxr-xr-x 7 root root 4096 Mar 6 2019 usr druxr-xr-x 11 root root 4096 Mar 6 2019 var With -d I run the container and it continues running , we use -d when we run for example a nginx service. To start a container.	Run Start
docker stop idOfCointaner.	To stop a container.	Stop
docker rm idOfCointaner.	To remove a container.	Remove
docker pull nameOfTheContainerImage	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 # ls /home # _	Run
docker exec -it IdOfTheContainer sh docker exec -it IdOfTheContainer bin/bash	To execute a container that it is running.	Execute

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 yout own image and run it: gaston@gaston:~\$ sudo docker run mydockerimage:1.0 figlet "My image has figlet"	
docker image Is docker images	In this example the image is taged with a name and version. To show our images: gaston@gaston:~\$ sudo docker image ls REPOSITORY TAG IMAGE ID CREATED SIZE nello-world latest bf756fb1ae65 5 months ago 13.3kB	Show
docker rmi idOfThelmage	alpine 3.7 6d1ef012b567 15 months ago 4.21MB gaston@gaston:~\$	Domovo
docker rmi idOfTheImage docker image tag IdOfTheImage	To remove an image. To tag the image, you can run docker image tag IdOfTheContainerImage myDockerImage:1.0 or docker image tag	Remove
docker image tag idominalimage	IdOfTheContainerImage myDockerImage tag IdOffneContainerImage myDockerImage:1.0 or docker Image tag IdOffneContainerImage myDockerImage:1.0 or docker Image tag IdoffneContainerImage myDockerImage:1.0 gaston@gaston:~\$ sudo docker image tag d24ea63ab15e mydockerimage:1.0 gaston@gaston:~\$ sudo docker image 1s 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:~\$	g

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 that 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 build cache Are you sure you want to continue? [y/N] y Deleted Containers: 44e7a2d5cd4554d5refa2eb94e3d16797fa5039ddd30230386affe5974e5197b 97757b94e680c7cc385e5d3c84c5d3c91d47135f531af1879e19665b445d609 d66e080c724c93aca8f2a2e5e48141d1def9222dde181dfaf2e2286fa58bd3aeb 64c00228d91lb3236284d50bd52954a4c0ece522a46e12070b075688cbe01164 1aebc5f890e73f918d246c6418abee31f7eb1bf270730ddd6a02a6367e9d3ed5c 5a146203fa525124b176924af64637a493abde137cc24d2b2cd11705bac29975 1c25e5c5eb4859fc7c505344031b133ab8d49b819a2681f3bbb8585fb61bab552 664adbdef87cfa6196727fcfdce4f2061037fd300e5bde04ac45b88aa110a 9f03fb060fbf439c2c7bdf8d87d14a2f8313fd6134ed20d7ebfd10a2c40f1200 f35e977547a72ac7ac3eald2fde6e762d8f1a70c5f392333c73e95bb60e0427d18 24ddd321170f596432ad9494f81ebba5dd97e884a7a4f922a3f5f34ad93d7bc c6191eb18bb6aac8d837003355e3d7d7160ad75bb6446bab0beop715b6df9b 045599ff725a72ac7ac3eald2fd26d64662d844f60ab0beop715b6df9b 045599ff725a73c595b4562c8be1858496cde0345d850ad3be1c308bef1a23c6 d060cc7357ddf61f3c242ce9ba3da625875193266104315479d4dfdaff9199c 0246775a6693954556c126f43bd49f375655c20d1707187a816a4517d366 0d1ba17f4ff6311e66f6bc7f289671031021e64b8cad2b60d985579924b83735 060224ae985c996d518650f46044875ee6483abbc6326e2c2d2bb6967902 47e489539980c26d6918630f60404875ee6483abbc6326e2c2d2bbd0e84716 66e56765df6aba7b5398f9fe96db6cf9d53e8ae4b46c7316fdd0b8f405a7ba 0148e5955aecccdeffde6392139826h173f588986f00ac1cd4dd6c7ba7ba 0148e5955aecccdeffde6392139826h173f588986f00ac1cd4ddb8f405a7ba 0148e5955aecccdeffde656c7f289671031021e64b8cad2b60d985579924b83735 060224ae985c96d5918690f64044875ee6483abbc6326e2c2d2b9bd0e82e2f1 66e56765df6aba7b5398f9fe96db6cf9d53eseae4b46c7316fdd0b8f405a7ba 0148e5955aecccdeffde666666679d53eseae4b46c7316fdd0b8f405a7ba 0148e5955aecccdeffde66666679d5986a6496600aecccd4ddc667055266606679d53eseaeb466c7316fdd0b8f405a7ba 0148e5955aeccccdeffde6666666674	
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 conta<u>iner to reference</u> 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
TUS
                   PORTS
d66e08072442
                      wordpress:5.4.2-php7.2-apache "docker-entrypoint.s.."
                                                                                          6 minutes ago
                                                                                                                  Цp
13 seconds
                   0.0.0.0:8080->80/tcp docker_wordpress_1
4c00228d911
                    mysq1:8.0.20
                                                              "docker–entrypoint.s…" 6 minutes ago
                   3306/tcp, 33060/tcp docker_mysql_1
18 seconds
gaston@gaston:~/docker$
```

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