**Docker Quick Guide**

**Hands-on Lab**

We are going to install ubuntu 16.04 image with docker in CentOS Linux release 7.9.2009 (Core)

**Objectives**

* Know the basic commands in Docker
* Install Ubuntu Image

**Steps**

1.- We check which images we have in or docker, in this case we only have the hello world image that we run when we install docker, so we need to pull the ubuntu image with the version 16.04. Something you need to know is when we pull some image maybe the image pulls other images for its own.

“docker pull ubuntu:16.04”

A screenshot of a computer

Description automatically generated

2.- To run an image we use the command run. We can specify de tag that we want, this tag usually represent the version of the image

“docker run ubuntu:16.04 or docker run [id]”

The ID of the image is truncate and we only see the last 12 characters, to see al id we use the command

“docker images --no-trunc”

A screenshot of a computer

Description automatically generated

3.- We are going to work on an existing image I made call onboarding, for this we need to use the user for docker and the name of the image

“docker pull [user]/[image name]:[version]”

By default if you don’t specify the version of the image the command will complete with “latest” so if you don’t put a version sometime you will have an error

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

4.- To create a new image we use the command and the next command is for run the container and name it, it means “interactive terminal” and its very important if you want to work in that container

“docker build [file name| .] ”

“docker container run -it [container] --name [name of the container]”

A screenshot of a computer

Description automatically generated

This name help us to identify the container we are running, we can see some random names on step 5 that’s because we don’t specify a name when we run the container, we only use the command

“docker run [container]”

A screenshot of a computer

Description automatically generated

5.- We can check if we have any container running with

“docker container ls -a”

Or we can check the processes with

“docker container ps -a”

Status (Life cycle)

Started, Running, Exited

For the lifecycle you can stop, start, attach and remove your containers

“docker [stop|start|attach|rm] [container]

You can’t use the attach command if you don’t start the container

A screenshot of a computer

Description automatically generated

6.- If you want to remove an image use

“docker rmi [image id]”

A screenshot of a computer

Description automatically generated

7.- If you want to save a new image first you login with the command

“docker login”

If your image doesn’t have repository and tag you can add this information with the command

“docker tag [image id] [user]/[repository name]:[tag]

The tag is usually use for the version of the container. After you have this information you can push your image to your docker hub with

“docker push [repository name]

8.- To update new features to an image first we are going to create a directory for save the docker file after that we are going to build or image and then push it

9.- To check which port is expose in our container we use the command

“docker image history [image]”

A screenshot of a computer

Description automatically generated

We run nginx in detach way, this container run in port 80. We can check with the previous command.

A screenshot of a computer

Description automatically generated

There is a difference between expose and publish. Expose opens and expose the port inside of the container and makes it available to local host and internal services. I can uses services like ELinks to connect to the port. On the other hand, we have publish that open both ports (host port and the container port) and the host port can be specified, and they are basically link to one another so that we have accessibility to the container form the outside world.

We can check the IP of the container with the command

“docker container inspect [container ID] | grep IPAdd”

And we can connect to that container with elinks, you can specify the port 80 but by default is already set. So you can use

“elinks [IP]:[Port(default 80)]”

In this case I specify the port anyway. We can check that nginx is working

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

To connect host port to container port we use the option (-p | -P) in the run command. -P choose a random port from the host. -p you specify which port you want to choose to connect with the container

“docker container run -p [Host port]:[Container Port Expose] [image] ”

“docker container run –P [image] ”

We are going to connect the port 80 from our host to 80 form the container and the way con test this configuration would be with elinks to our host port

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA black screen with white text

Description automatically generated

Docker volumes are the preferred mechanism for persisting data that’s generate by containers, has several advantages like we can use in Linux and Windows containers.

You can use volume drivers to let you store volumes on a remote host or even a cloud provider. To check volumes use the command

“docker volumes ls”

To create a volume use

“docker volume create [volume name]”

To inspect volume use

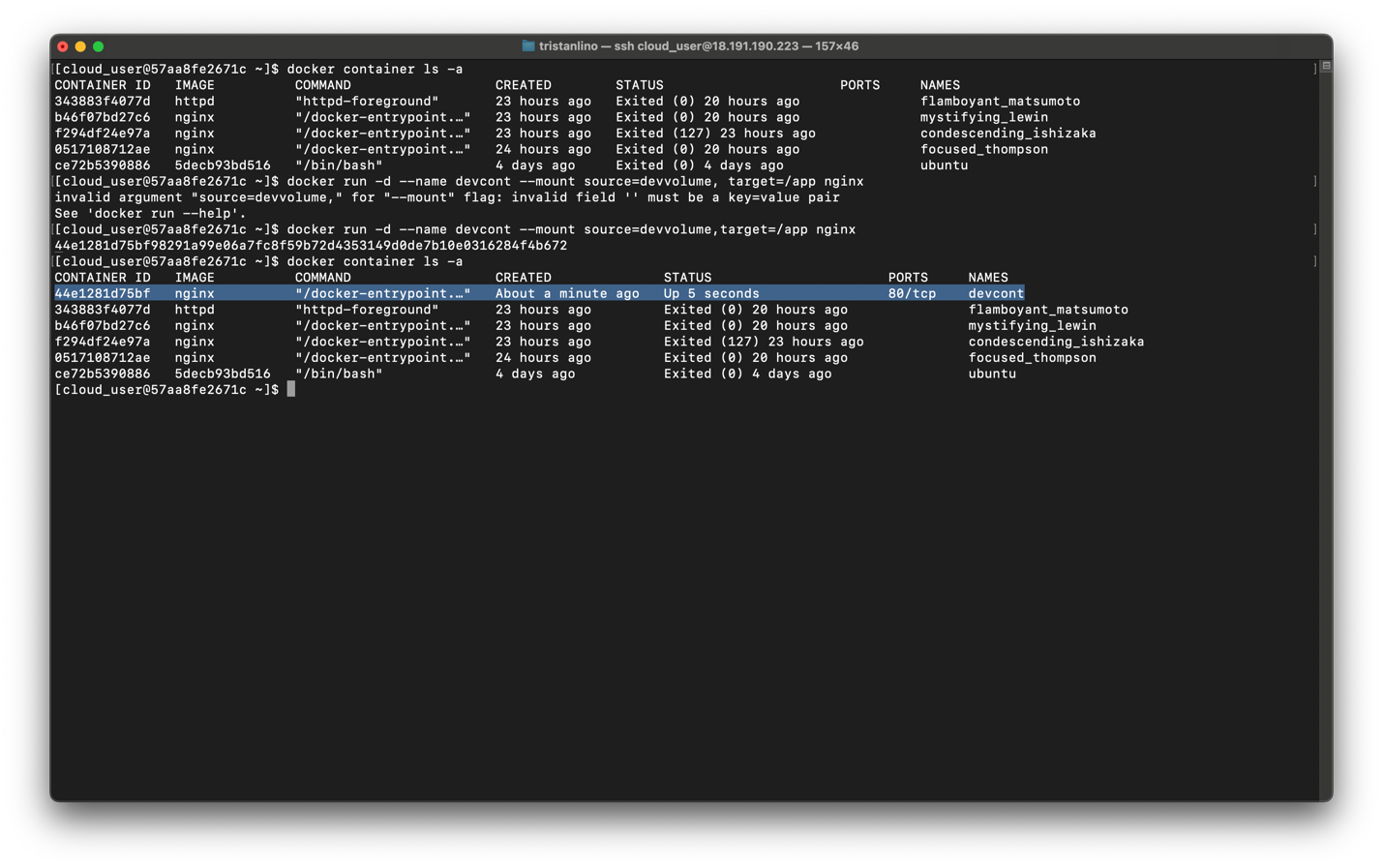
“docker volume inspect [volume name]”

A screenshot of a computer

Description automatically generated

To mount a volume we use the command --mount or -v

“docker container run -d --name [name container] --mount source=[volume name],target=/app [image name]”



We can check if the container has the volume mount or attach with the command

“docker container inspect [container name]”

And search the word “Mount”

A screenshot of a computer

Description automatically generated

We can check the data inside of the volume with

“sudo ls [path from Source]”

To open an interactive terminal we use the exec command with the -it option

“docker container exec -it [container name] sh”

A screenshot of a computer

Description automatically generatedEven if we delete de container the data still remained in the volume. We can attach this data to any volume we want

A screenshot of a computer

Description automatically generated