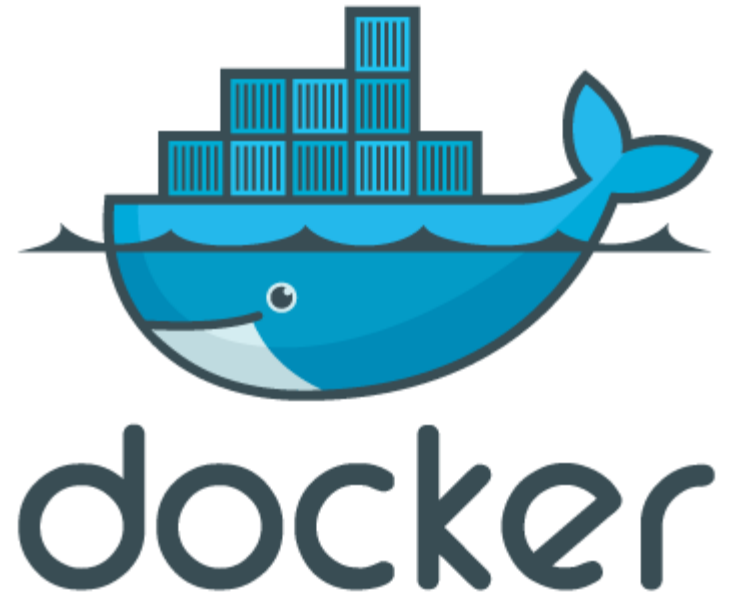


Docker

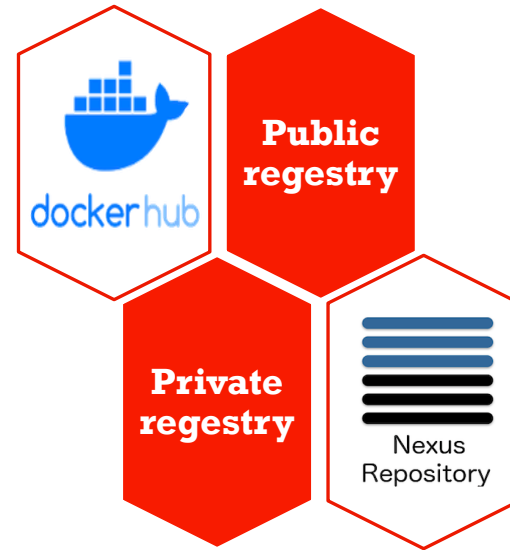
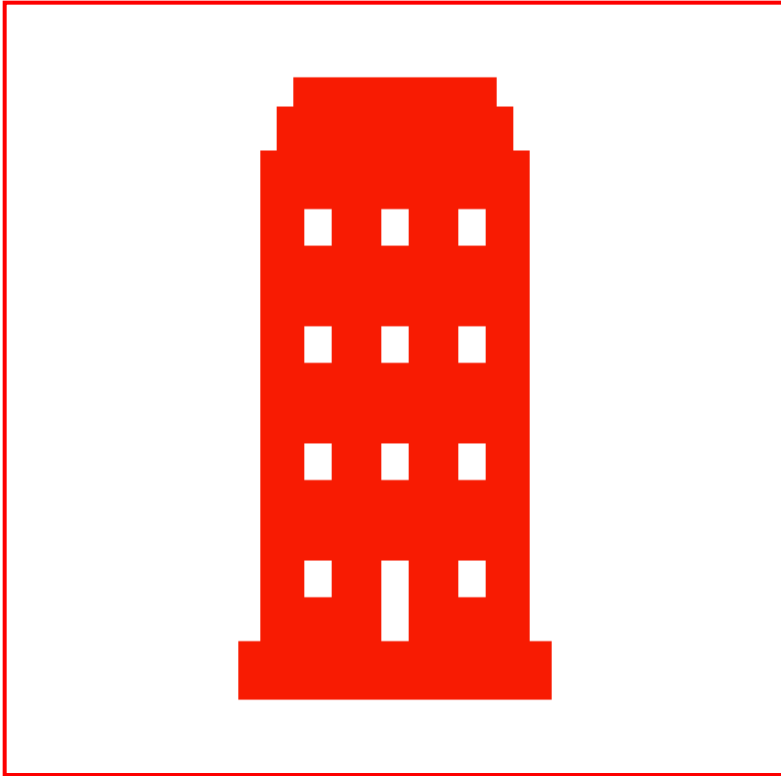
Presented by:
Jospheen Boles



Outline

- **Containerization VS Virtualization (Container vs VM)**
- **What's Docker and its architecture?**
- **Docker images**
- **Docker Container**
- **Docker registry**
- **Docker components (Client, Host and Daemon)**
- **Common docker commands**
- **Dockerfile and its instructions**

Types of Docker registries





Manipulating Containers with the Docker Client

Docker Basic Commands

Build Image

- `$ docker build -f Dockerfile -t <image-name:tag-name>`

Overriding default command

- `$ docker run <image-name> <command!>`

Attached and Detached

- `$ docker run -d <image-name>`
- `$ docker attach <container ID>`

Interactive mode

- `$ docker run -i <image-name>`

Port mapping

- `$ docker run -p host:container <image-name>`

Docker Basic Commands

Volume mapping

- `$ docker run -v host:container <image-name>`

Inspect container

- `$ docker inspect <container-name>`

Stop container

- `$ docker stop CONTAINER`

List all images

- `$ docker images -a`

List all containers

- `$ docker ps -a`

- docker build

Creates Docker images from a Dockerfile

A terminal window with a dark background. The title bar at the top says "Docker build" and has standard window control buttons (minimize, maximize, close). The command prompt shows a single line of code: `1 docker build -f Dockerfile -t images_name:tag .`

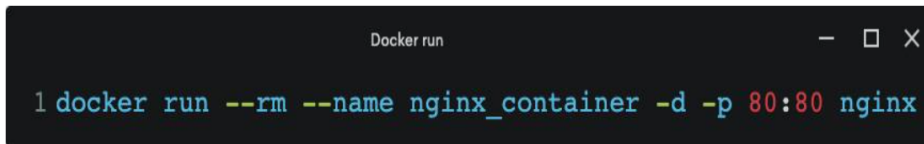
```
Docker build
1 docker build -f Dockerfile -t images_name:tag .
```

Detail:

- `-f` : Specifies the name or location of the Dockerfile
- `-t` : Assigns the name of the image with its respective tag

- docker run

Allows you to run a container from an image

A terminal window with a dark background and light gray window controls (minimize, maximize, close) in the top right corner. The title bar reads "Docker run". The command prompt shows a single line of code: `1 docker run --rm --name nginx_container -d -p 80:80 nginx`. The text is color-coded: `1` is light blue, `docker run` is green, `--rm` is yellow, `--name` is light blue, `nginx_container` is light blue, `-d` is green, `-p` is red, `80:80` is red, and `nginx` is light blue.

```
Docker run
1 docker run --rm --name nginx_container -d -p 80:80 nginx
```

Detail:

- `--rm` : Allows the container to be removed once it stops.
- `--name` : Allows you to assign a name to a container
- `-d` : Allows to run a container in detached mode
- `-p` : Allows ports to be exposed to the host interface

- docker container ls

Allows to list containers

A terminal window with a dark background. The title bar at the top says "Docker container" and has standard window control buttons (minimize, maximize, close). The command "1 docker container ls --all --quiet" is entered in a light blue monospace font.

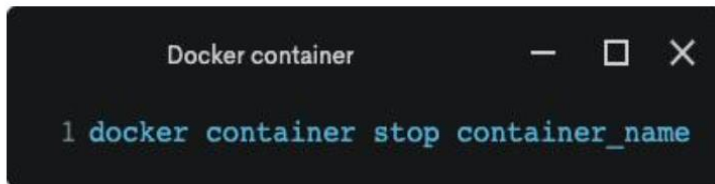
```
1 docker container ls --all --quiet
```

Detalle:

- --all, -a : List all containers up to the detainees
- --quiet, -q : Lists container IDs only

- docker container stop

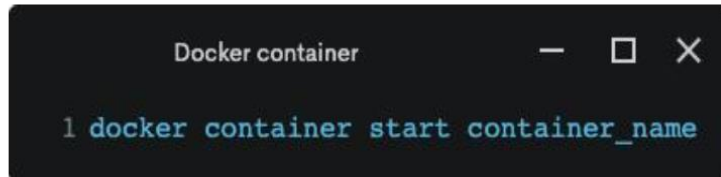
Allows to stop a container

A terminal window with a dark background. The title bar at the top reads "Docker container" and has standard window control buttons (minimize, maximize, close). The terminal contains a single line of text: "1 docker container stop container_name".

```
Docker container — □ ×  
1 docker container stop container_name
```

- docker container start

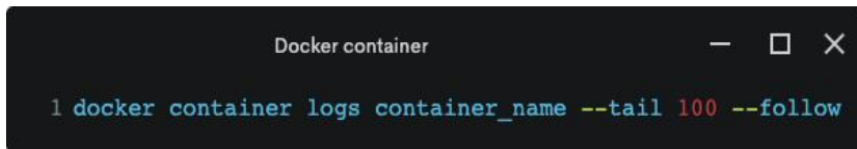
Allows you to start a container

A terminal window with a dark background. The title bar at the top reads "Docker container" and has standard window control buttons (minimize, maximize, close). The terminal contains a single line of text: "1 docker container start container_name".

```
Docker container — □ ×  
1 docker container start container_name
```

- docker container logs

Allows you to view the logs of a container

A terminal window with a dark background. The title bar at the top says "Docker container" and has standard window control buttons (minimize, maximize, close). The command prompt shows a single line of code: `1 docker container logs container_name --tail 100 --follow`. The text is color-coded: "1" is blue, "docker" is green, "container" is green, "logs" is green, "container_name" is green, "--tail" is green, "100" is red, and "--follow" is green.

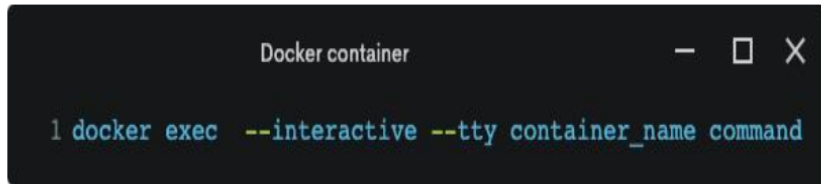
```
Docker container
1 docker container logs container_name --tail 100 --follow
```

Detalle:

- `--tail, -n` : Number of lines to display
- `--follow, -f` : Keeps track of the output being generated

- docker container exec

Allows to execute a command in a running container

A terminal window with a dark background. The title bar reads "Docker container" and has standard window controls (minimize, maximize, close). The command prompt shows a single line of text: `1 docker exec --interactive --tty container_name command`.

```
Docker container
1 docker exec --interactive --tty container_name command
```

Detalle:

- `--interactive, -i` : Connect to STDIN
- `--tty, -t` : Assign pseudo-TTY terminal

- docker stats

The docker stats command returns container consumption statistics such as memory, cpu and networking usage.

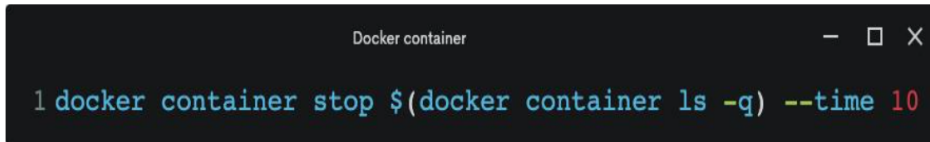
```
CONTAINER ID   NAME          CPU %       MEM USAGE / LIMIT   MEM %      NET I/O       BLOCK I/O    PIDS
02017f310c3f   sad_vaughan   0.00%      3.793MiB / 1.944GiB  0.19%     1.66kB / 0B   205kB / 0B   5
```

Detalle:

- --all -a: Displays statistics of all running containers.

- Concatenated commands

This CLI allows concatenated commands to simplify management tasks.

A terminal window titled "Docker container" with standard window controls (minimize, maximize, close). The prompt is "1" and the command entered is "docker container stop \$(docker container ls -q) --time 10". The command is color-coded: "1" is blue, "docker" is green, "container" is green, "stop" is green, "\$(" is green, "docker" is green, "container" is green, "ls" is green, "-q)" is green, "--time" is green, and "10" is red.

```
1 docker container stop $(docker container ls -q) --time 10
```

Details:

- First, the command found in `$()` is executed, which outputs the container IDs
- Subsequently a `docker container stop` will be executed for each of the IDs with the `--time` flag to stop after 10 minutes.

Lab2

Problem 1

- Run the container hello-world
- Check the container status
- Start the stopped container
- Remove the container
- Remove the image

Problem 2

- Run container centos in an interactive mode
- Run the following command in the container “echo docker ”
- Open a bash shell in the container and touch a file named hello-docker
- Stop the container and remove it. Write your comment about the file hello-docker
- Remove all stopped containers

Problem 4

- Run the image httpd again without attaching any volumes
- Add html static files to the container and make sure they are accessible
- Commit the container with image name IMAGE_NAME
- Push it to docker hub