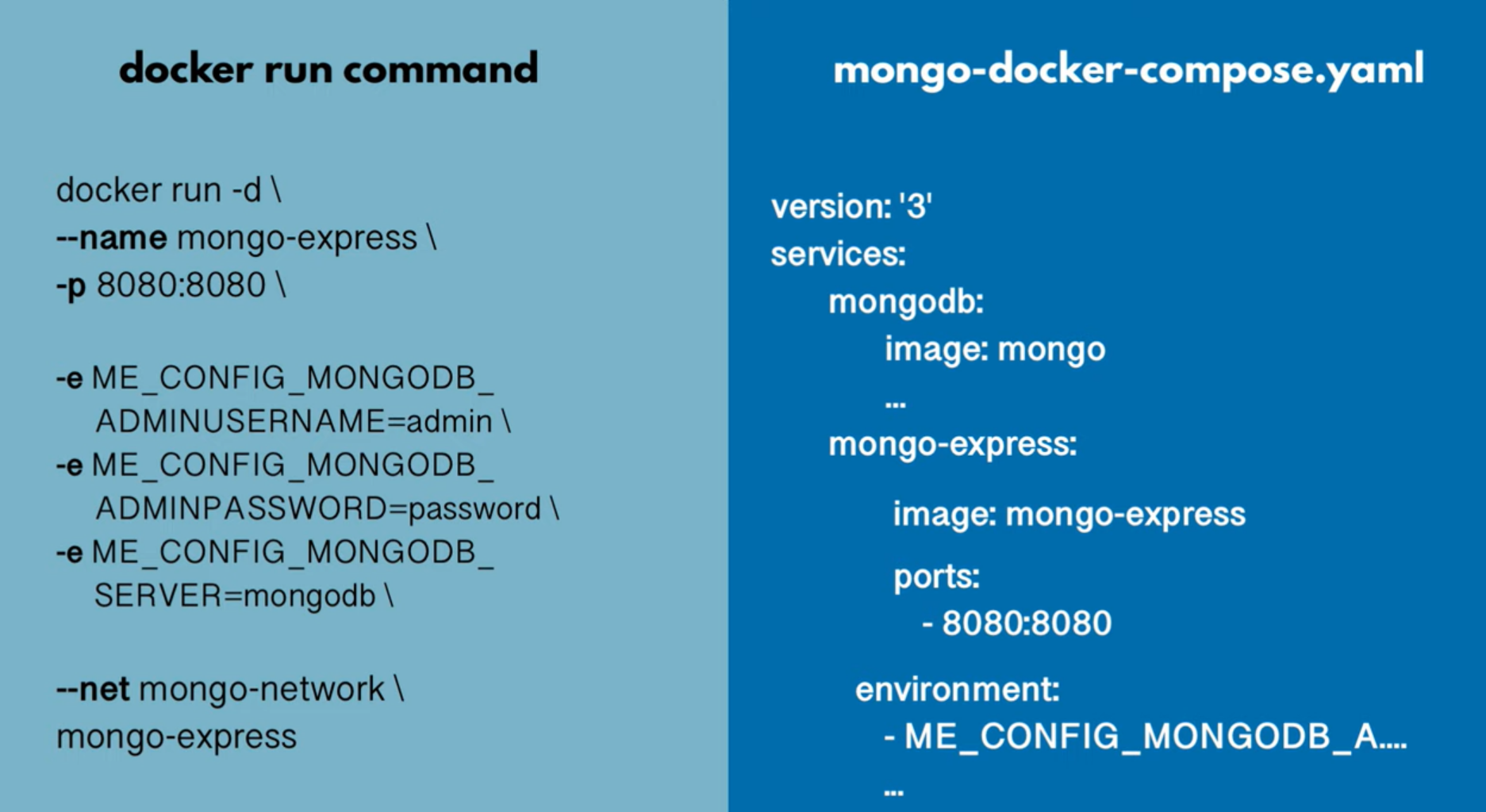
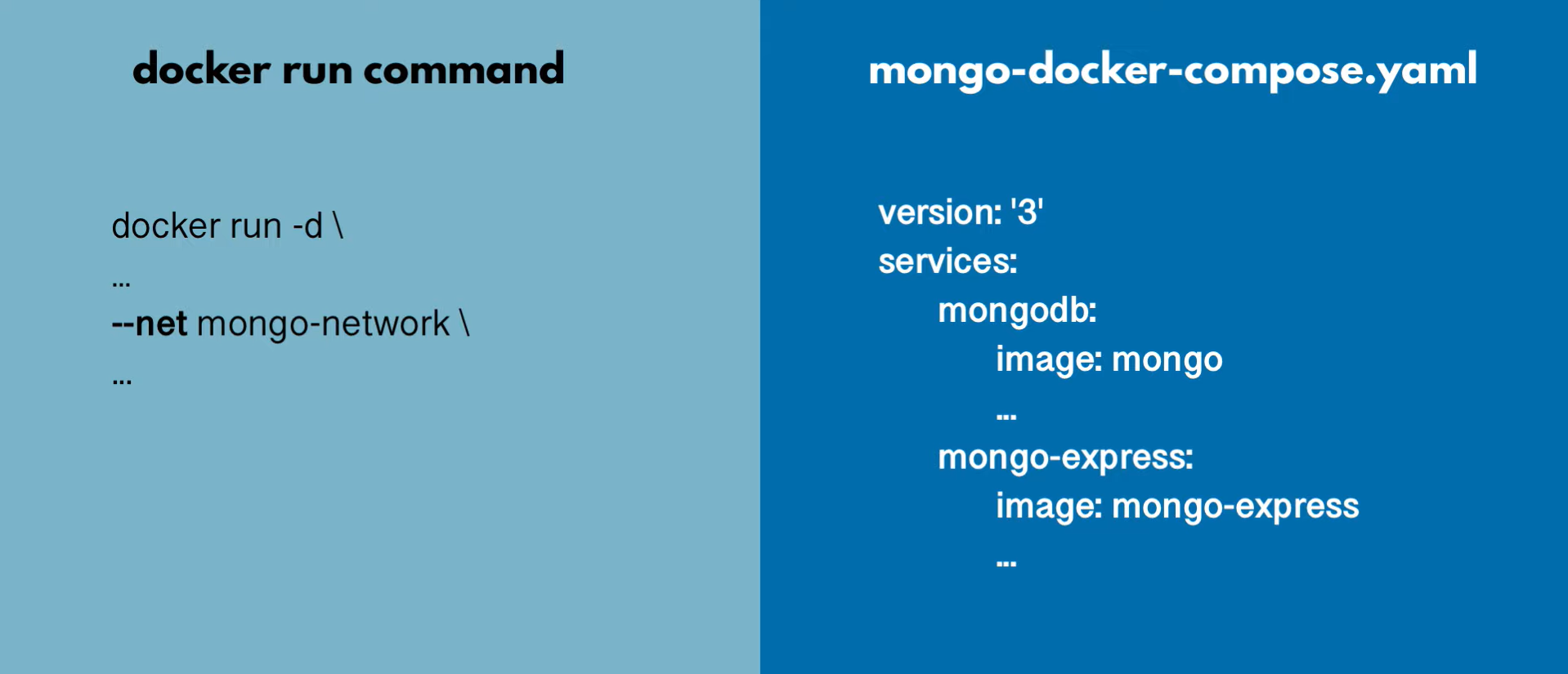
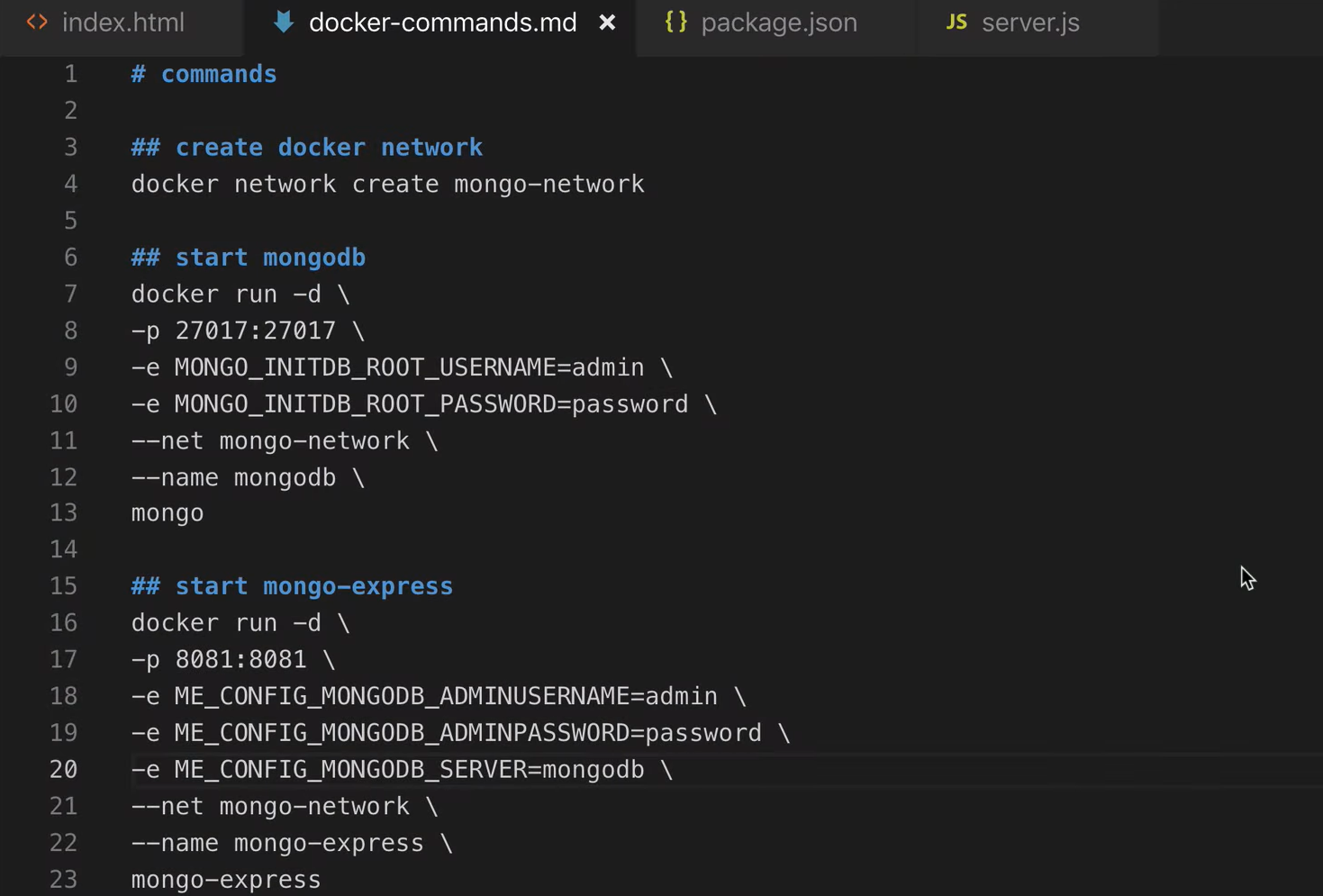
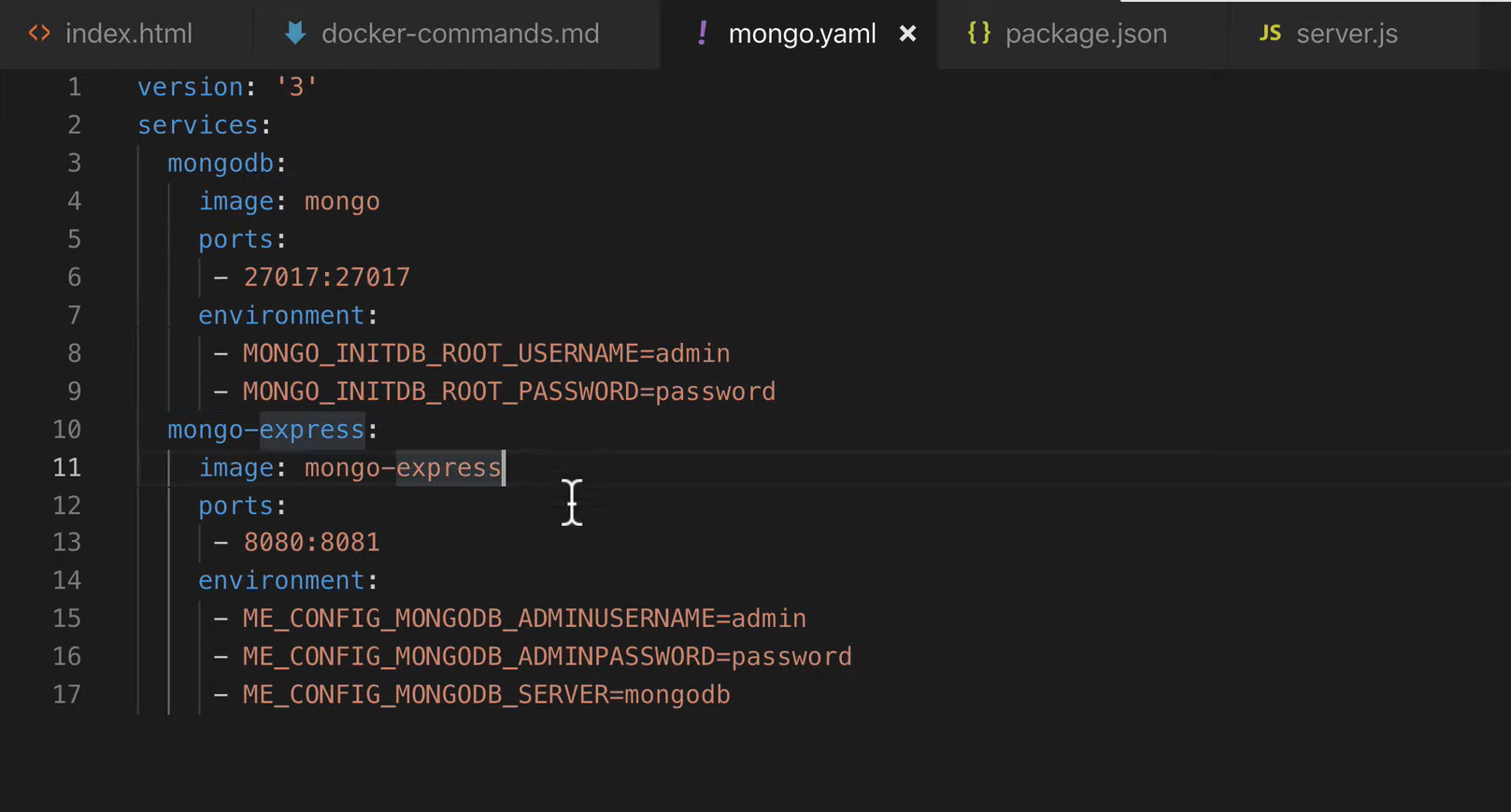
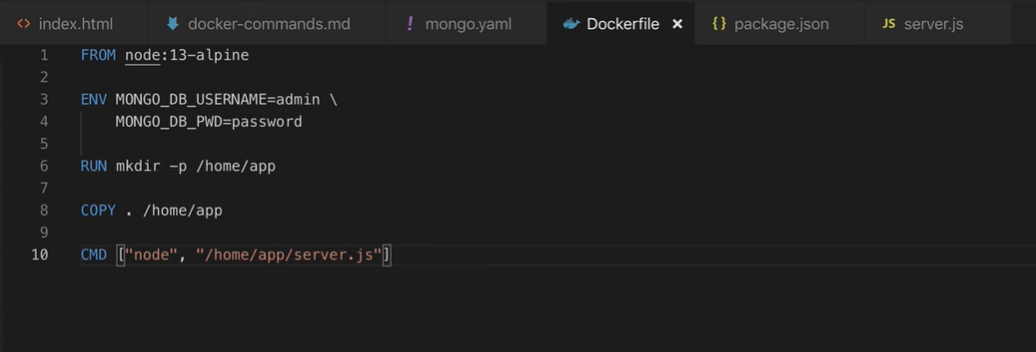
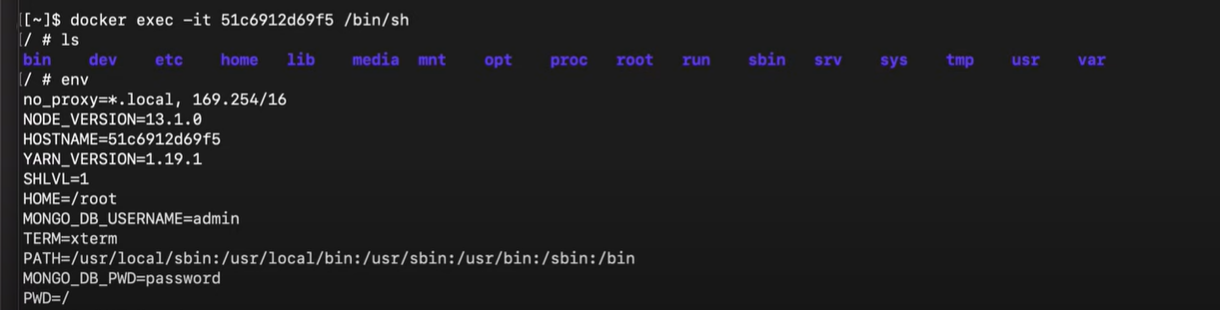
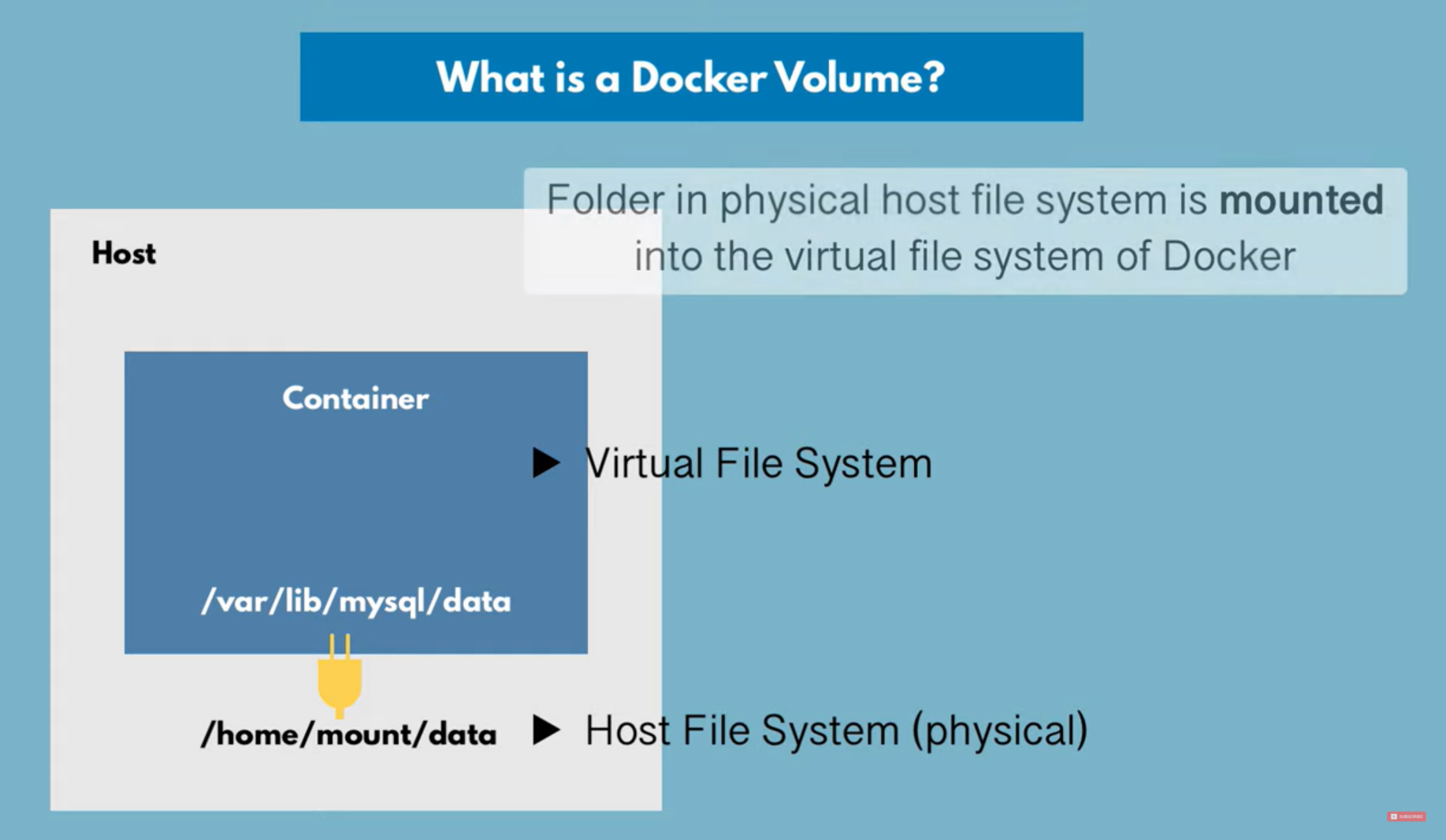
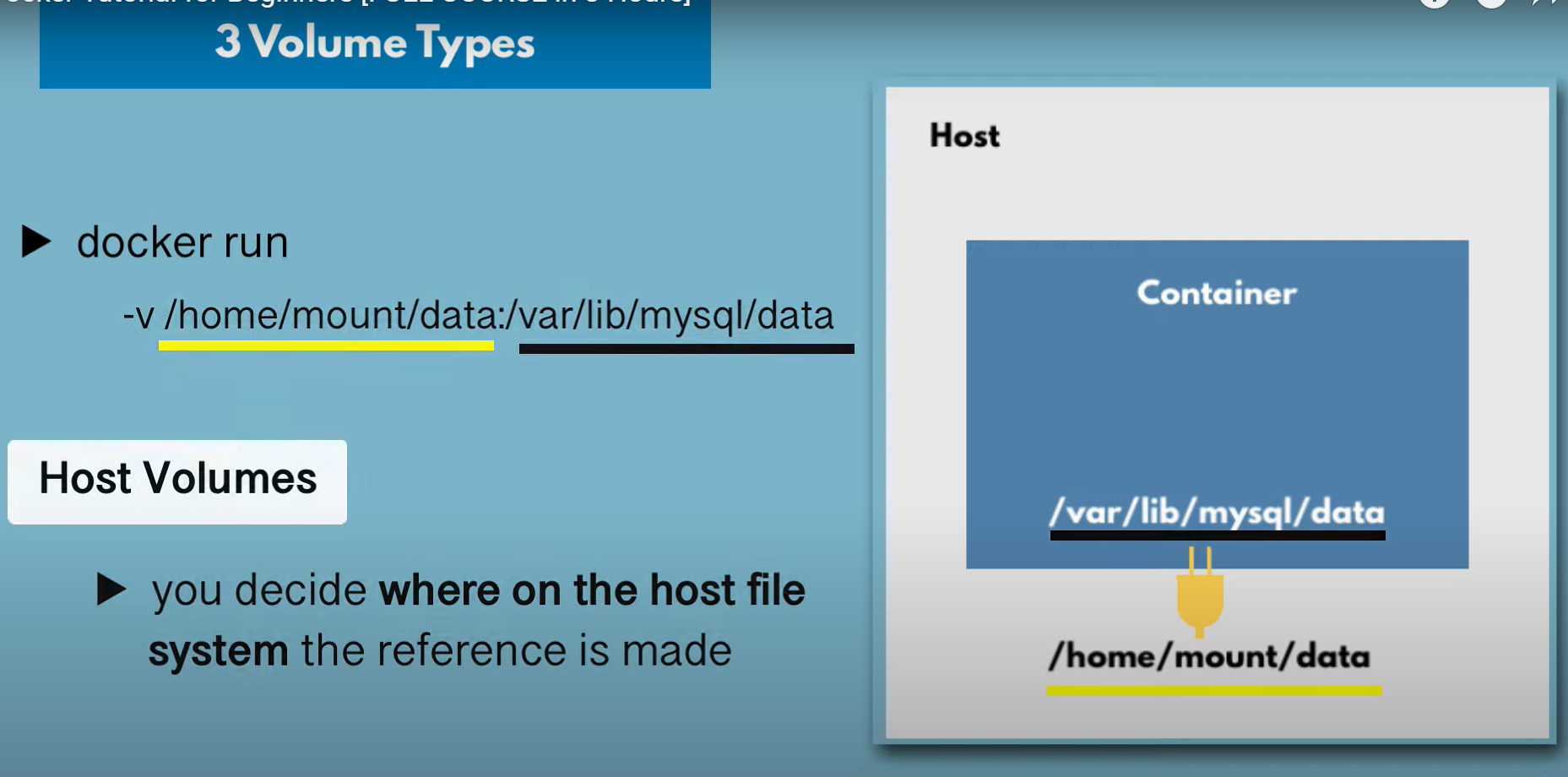
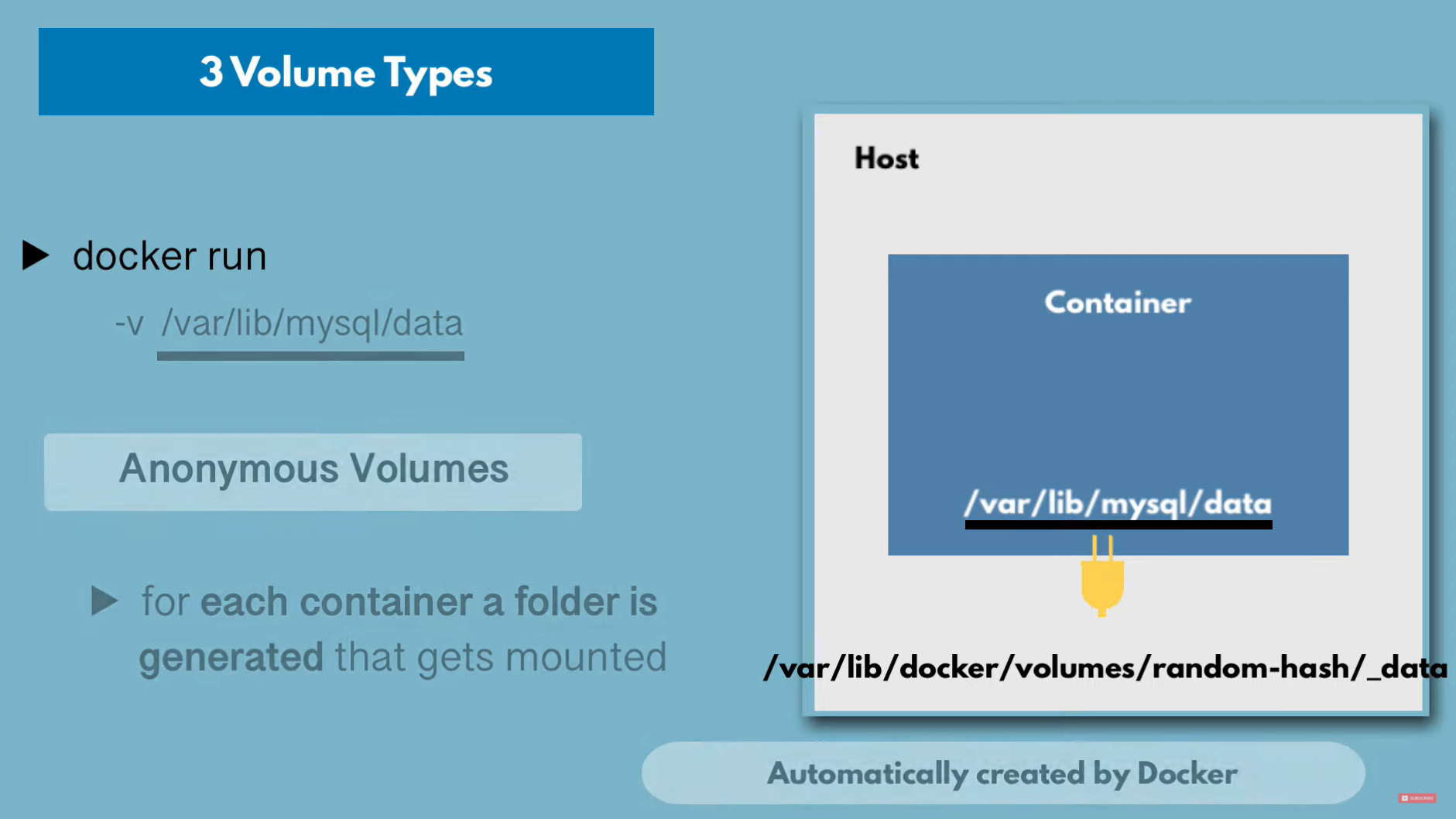
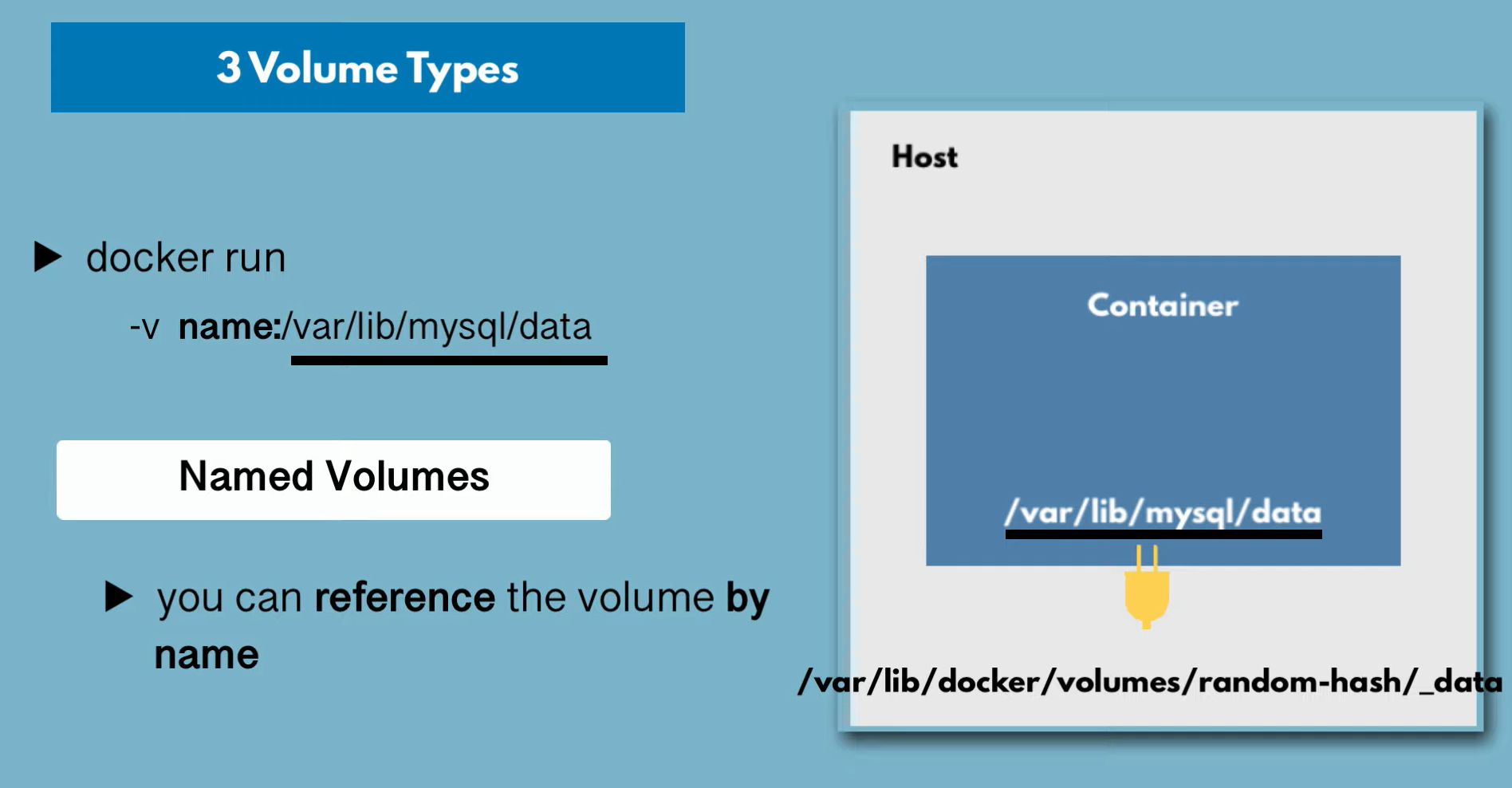
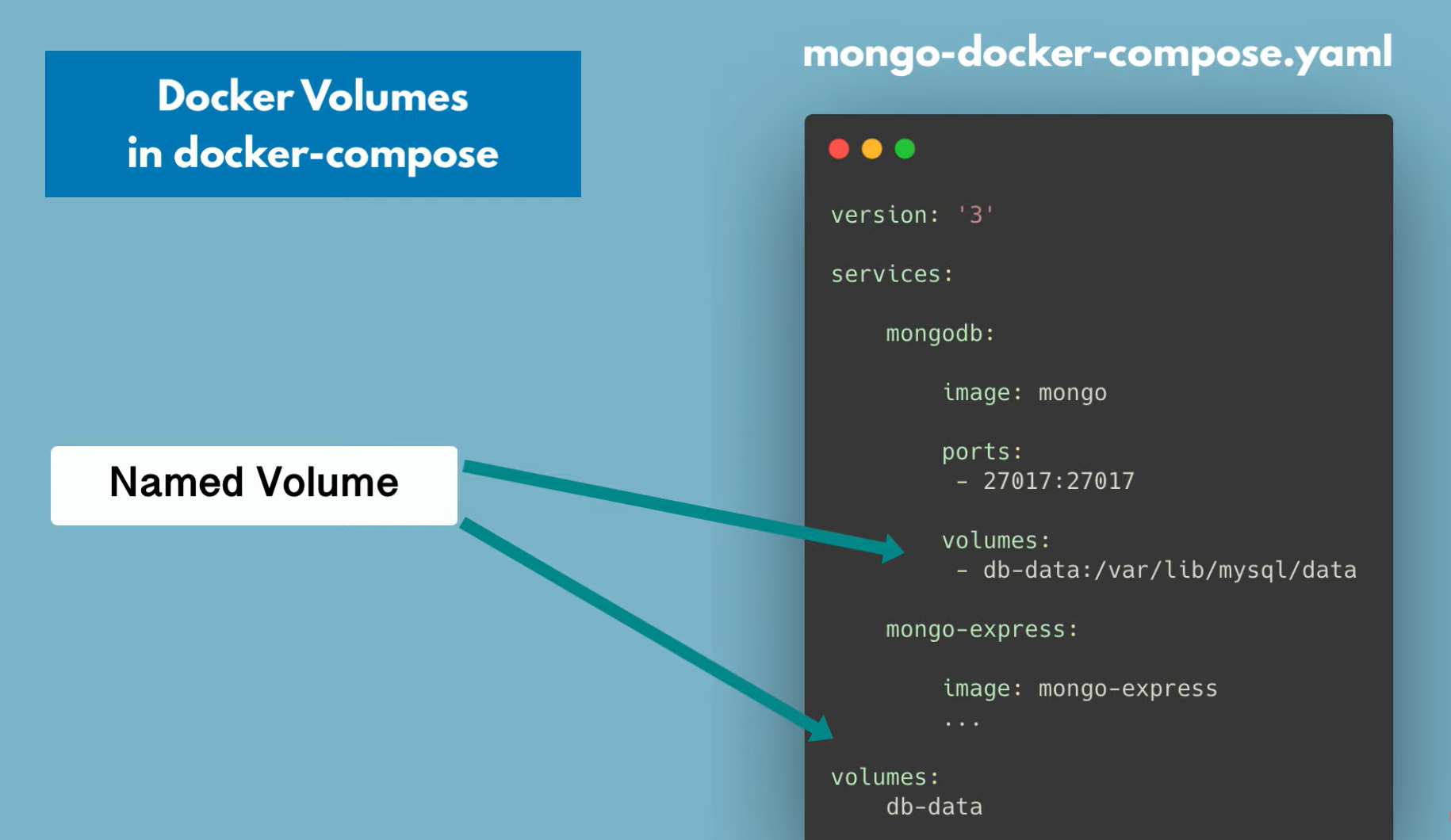
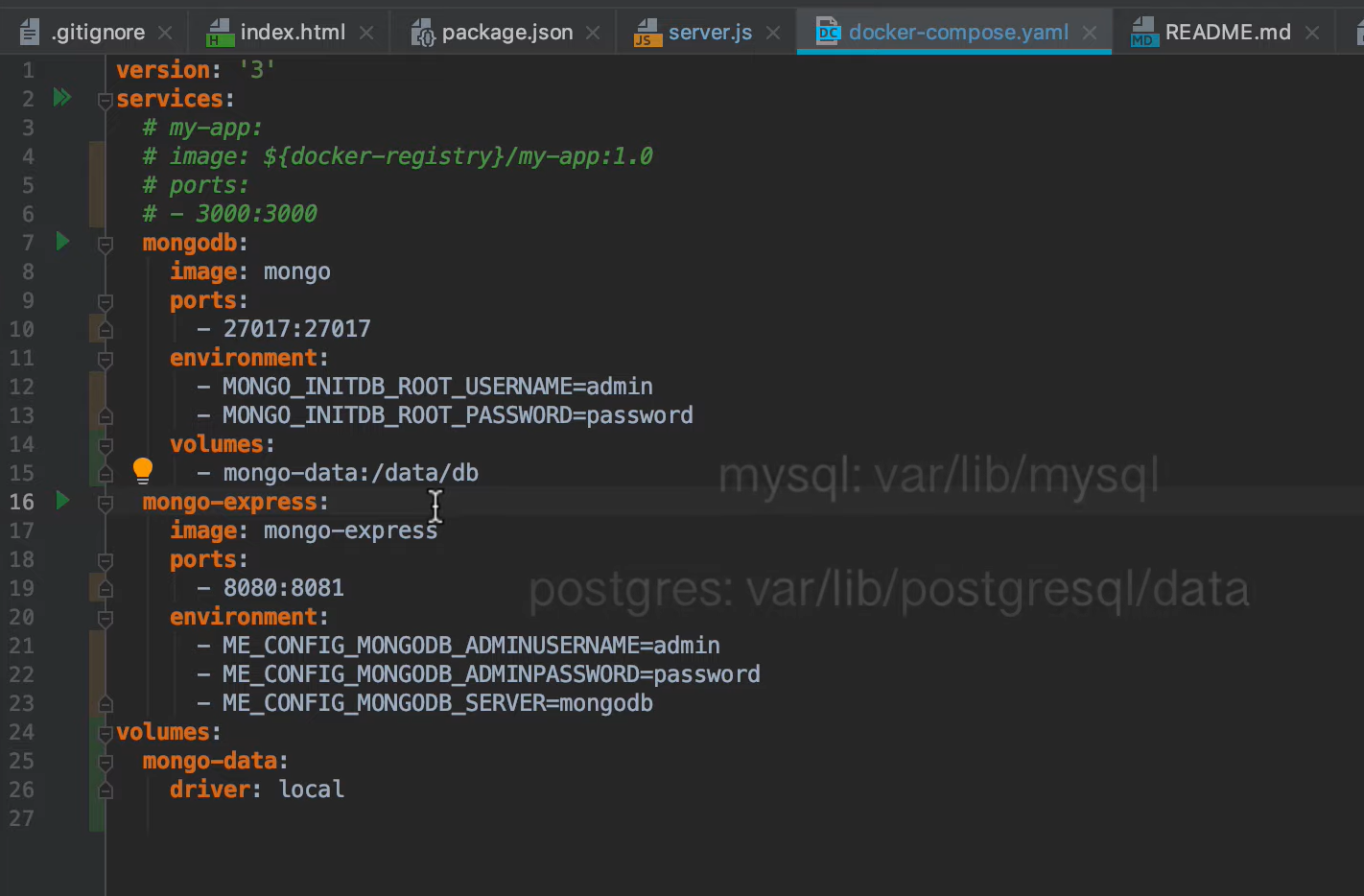
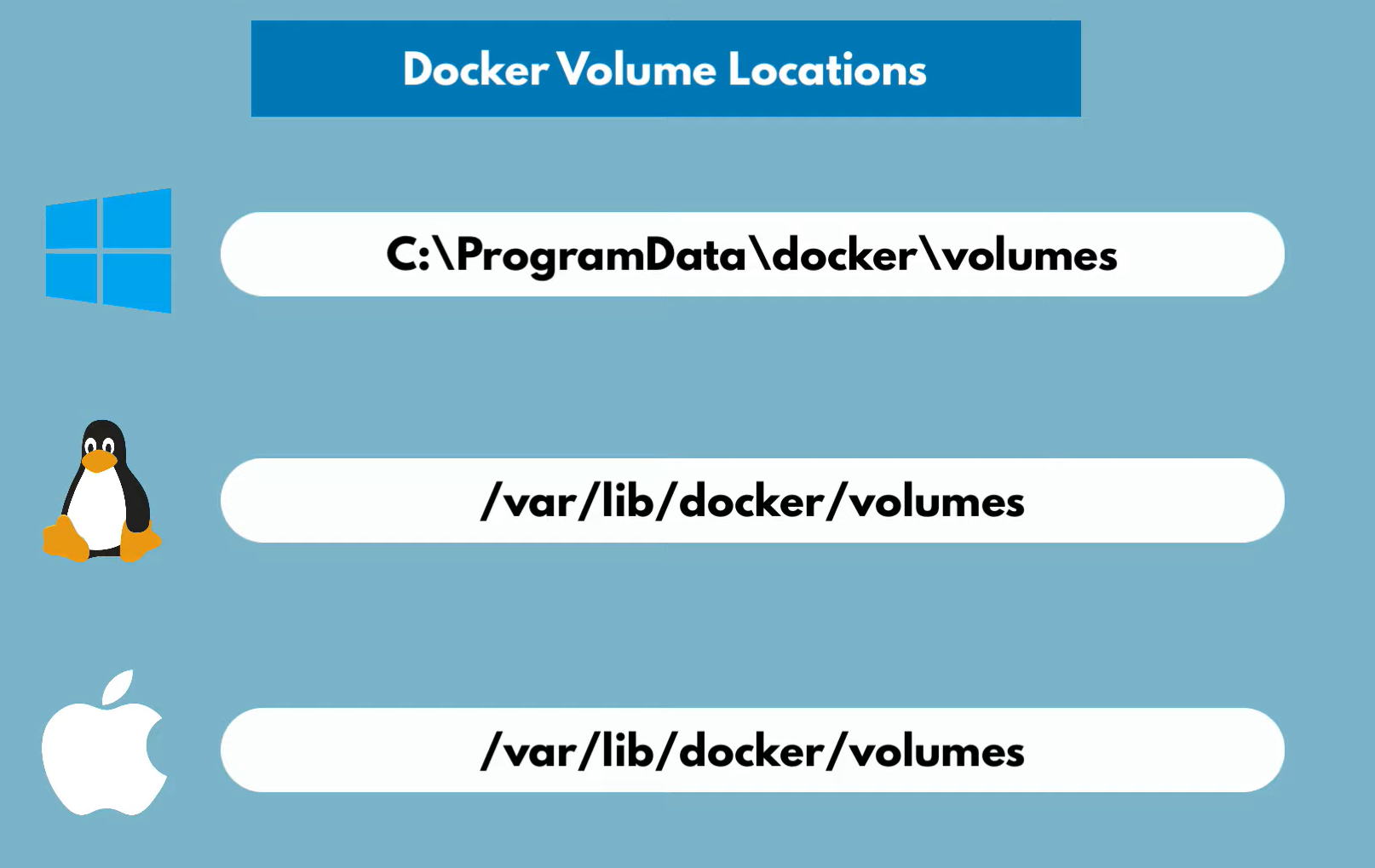
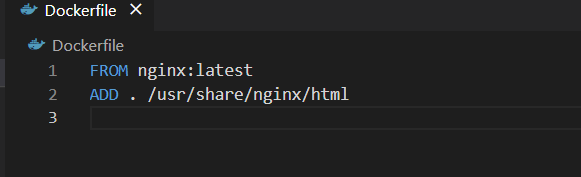
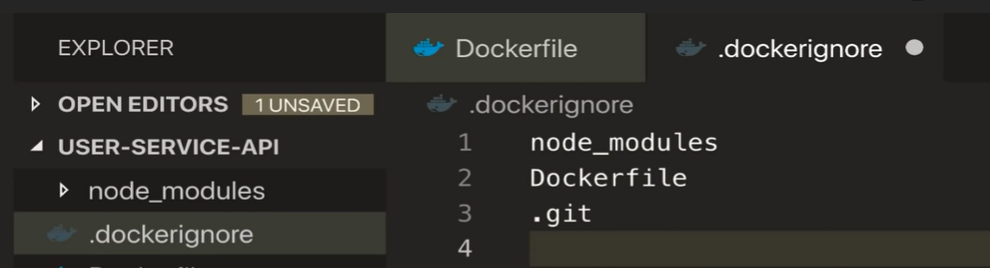
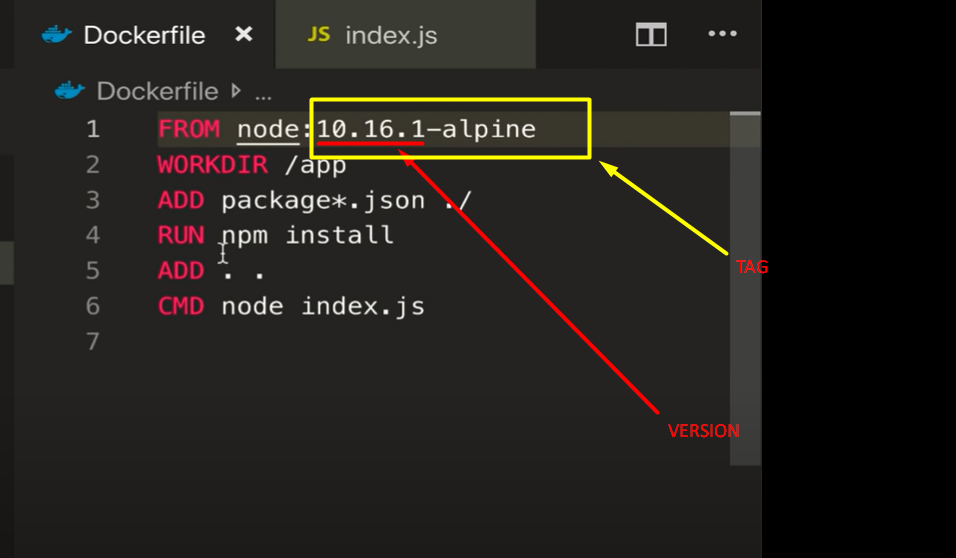
* docker ps = prints all running containers
* docker ps -a =print all containers
* export FORMAT="ID\t{{.ID}}\nNAME\t{{.Names}}\nIMAGE\t{{.Image}}\nPORTS\t{{.Ports}}\nCOMMAND\t{{.Command}}\nCREATED\t{{.CreatedAt}}\nSTATUS\t{{.Status}}\n"
* // usage:
* docker ps --format="$FORMAT"
* docker ps -a -q / docker ps -aq = print all containers ids
* docker container ls = print all running containers
* docker container ls -a -q = print all containers ids
* docker images = print all images
* docker run <image\_id / image\_name> = create and run a container
* docker stop <container\_id / container\_name> = stop the running container
* docker start <container\_id / container\_name> = start container
* docker rm <container\_id / container\_name> = remove container
* docker rmi <image\_id / image\_name> = remove image
* docker rm ${docker ps -a -q} = remove all containers
* docker run -d = run container in detached mode
* docker logs <container\_id / container\_name> = show logs from container
* docker attach <container\_id / container\_name> = attach to running container
* docker run --name my\_container\_name = create and start a container with specified name
* docker exec -it <container\_id / container\_name> /bin/bash = lunch interactive terminal in specified container
* docker exec -it <container\_id / container\_name> /bin/sh = lunch interactive terminal in specified container
* docker exec -it <container\_id / container\_name> bash = lunch interactive terminal in specified container
* docker network create my\_containers\_network = create a docker network
* docker run --net my\_containers\_network <image\_id / image\_name> = run container in network
* docker run -e ROOT\_USERNAME=someusername -e ROOT\_PASSWORD=somepassword <image\_id / image\_name> = run container with specified environment variables
* docker run -p host\_port:container\_port <image\_id / image\_name>
* 
* 
* 
* 
* docker-compose -f mongo.yaml up = execute mongo.yaml file
* docker-compose -f mongo.yaml down = stop all containers from mongo.yaml file
* 
* docker build -f Dockerfile -t my\_app:1.0 . = build a docker image based on Dockerfile from current directory
* 
* 
* 
* 
* 
* Last type of volume is Named Volume which is the most used!!!
* 
* 
* 
* docker run -d -p 8080:80 --name website -v C:\Users\relus\Personal\SWD\Repositories\Github\website:/usr/share/nginx/html:ro nginx = creaza un container cu un volum mapat
* Docker run --name copy-website --volumes-from website -d -p 8081:80 nginx = creaza un container al carui volum pointeaza catre volumul container-ului website
* **Dockerfile** Commands:
  + 
  + 
  + RUN vs CMD:
    - **RUN**: Can be many, and it is used in **build** process, e.g. install multiple libraries
    - **CMD**: Can only have 1, which is your **execute** start point (e.g. ["npm", "start"], ["node", "app.js"])
* **.dockerignore**
  + 
* **Caching**
  + 
* **TAG & VERSIONING**
  + 
* **Registries**
  + 