

# Homework 4

## Find the primary difference between Docker and Podman

The key difference between Docker and Podman lies in architectural design:

- Docker's design is a client-server-based design. It uses a daemon, an ongoing program running in the background, to create images and run containers.
- Podman, on the other hand, directly executes and runs instructions on the system without the need for a daemon to manage the containers.

## Document how to create, build and run a container with a Dockerfile

Once you have Docker installed, Create a Dockerfile with the instructions to build the image. As an example, there is a Dockerfile for a NodeJs app using Typescript in this folder.

```
PS C:\Users\PC\Desktop\Homework 4> docker build -t nodejs-in-docker .
[+] Building 1.1s (12/12) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 667B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/node:16-slim                  0.6s
=> [internal] load build context                                                  0.5s
=> => transferring context: 480.97kB                                             0.5s
=> [1/7] FROM docker.io/library/node:16-slim@sha256:197c40ee61039103188b3a96046fb8c90fd09b60f44c22a080bbe70b87c 0.0s
=> CACHED [2/7] RUN mkdir -p /home/node/app                                     0.0s
=> CACHED [3/7] WORKDIR /home/node/app                                          0.0s
=> CACHED [4/7] COPY --chown=node package*.json ./                             0.0s
=> CACHED [5/7] RUN npm install                                                  0.0s
=> CACHED [6/7] COPY --chown=node . .                                           0.0s
=> CACHED [7/7] RUN npm run build                                                0.0s
=> exporting to image                                                            0.0s
=> => exporting layers                                                            0.0s
=> => writing image sha256:26629e3221ada97b131037ae276cea7d41b156e57e53275fedd46a7466f649b2 0.0s
=> => naming to docker.io/library/nodejs-in-docker                             0.0s
PS C:\Users\PC\Desktop\Homework 4> docker run --name nodejs-containe --env-file .env nodejs-in-docker
```

i docker run --name nodejs-containe --env-file .env nodejs-in-docker

