Objective

The objective of this workshop is to

a. Build a Docker image
b. Deploy 2 containers inside Docker network

Setup

a. Create an account at Docker Hub (https://hub.docker.com)
b. Login to your Docker Hub account
c. Clone the repository https://github.com/stackupiss/cfdsa.git.

Workshop

Task 1

Create a directory called `workshop01/task1` in your repo. Unzip the given `dov-bear` file inside this directory.

Go into the the `dov-bear` directory. In this directory you will find 4 directories each with a same web application written in 4 different languages (JavaScript, Java, Golang, Python).

The web application randomly displays images on its landing page on port 3000. You can optionally display 2 message above the image.

See image (Figure 1) on the right.

The 2 messages and the port can be set with the following environment variables

- `PORT` - sets the port. Defaults to 3000
- `INSTANCE_NAME` - sets the application name. Will be displayed above the image. Defaults to empty string
- `INSTANCE_HASH` - sets an arbitrary hash for the application. Defaults to an empty string.

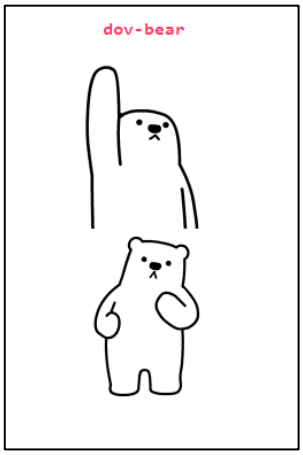


Figure 1 dov-bear

Select a language you are familiar with and containerize the application.

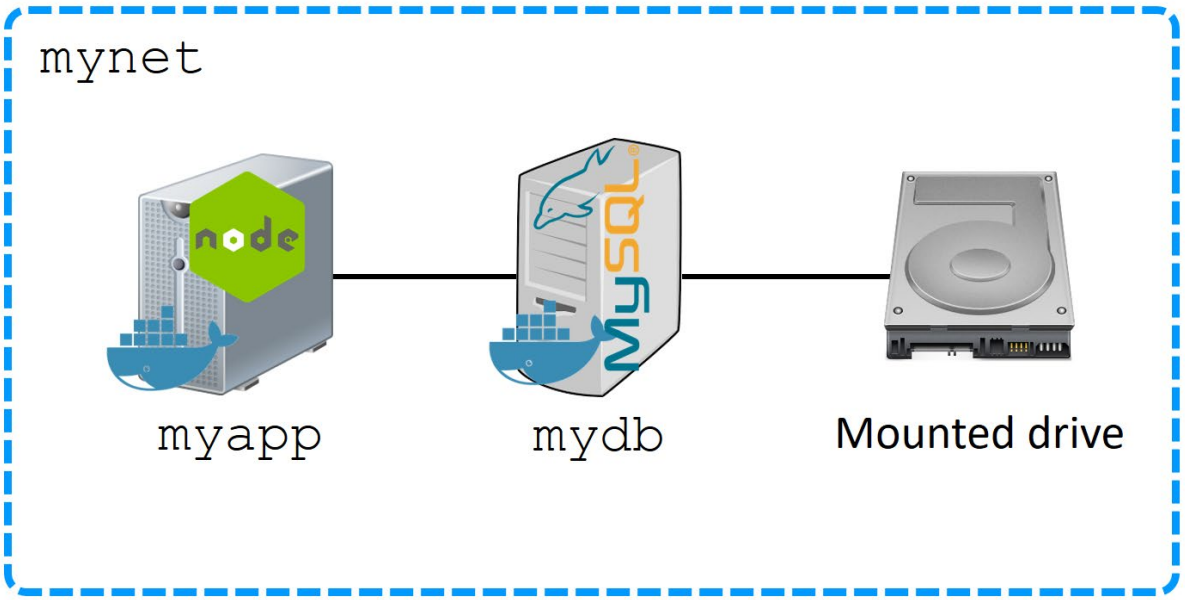
Once you have containerized the application push it to your Docker Hub account.

Run the image, set the environment variable `PORT` to 5000, `INSTANCE_NAME` and `INSTANCE_HASH` to any string value of your choice.

Task 2

Create a separate directory called `workshop01/task2` for the following.

Create the following Docker network (`mynet`)



The application consists of the following 2 parts

a. `myapp` – this is a Node based web application. It connects to a database; the database configuration is as follows
   i. `DB_HOST` is the database server
   ii. `DB_USER` is `root`
   iii. `DB_PASSWORD` is `changeit`

   Use the image `stackupiss/northwind-app:v1`

b. `mydb` – this is the database that your application will be using. Use the following image `stackupiss/northwind-db:v1`
c. Deploy the 2 containers inside a network called `mynet`.

   Don't forget to set the `DB_HOST` environment variable to the Node application (`myapp`) container.

d. In order to preserve the database when `mydb` is deleted, you should create a volume for `mydb` to save the database.

   Note that `mydb` will create a new database whenever `mydb` image is started viz. when a new container is created from that image

e. Deploy your application and access it with the Docker host IP address

Create a file called `steps.txt` and record the Docker commands you used to deploy this application

Optional Workshop

Only attempt this if you have completed the workshop.

Use Docker compose to deploy the above environment.

Submission

Push your work to your remote repository.

Post your Git repo on the submission channel in the course's Slack channel