

# Docker.

## Basics docker exercises

### Before to start...

- Check the section 1, 2 and 3 from the Udemy's docker course <https://www.udemy.com/docker-10days>

Please only focus on seeing the videos, you will have more time to practice after finish them.

These are some proposal for exercises that will help you to understand the basics of docker, but as we know, you have a curious mind, so don't feel like these are the only exercises that can help you, you can do more if you want.

- Your first **docker run**.
  - Install docker and execute the following command, **docker run -it ubuntu**
  - Play with that container for 15 minutes, you can create files, look directories and try to understand better what is a docker container.
- Understanding the basics command of docker.
  - Docker ps ,docker ps -a
  - Docker images
  - Docker inspect
  - Docker logs
- Environment variables
  - Bring up a mongodb container and set your own credentials via environment variables.
- Daemonizing your first container.
  - Understand what is daemonizing a container.
- Volumes and port settings.
  - Bring an nginx container, and mount your own directory of statics file, and make it available at port 80.

**If you are sure that you know all of the above things, speak with your manager, he will evaluate you in 10 minutes, so you can move forward to the following section.**

# Intermediate docker exercises

## Before to start...

- Check the section 5 and 6 from the Udemy's docker course <https://www.udemy.com/docker-10days>

Please only focus on seeing the videos, you will have more time to practice after finish them.

- Writing your first Dockerfile.
  - Create a dockerfile from alpine base image and install and configure awscli. The proposal for this container is to run awscli commands, after that it should die. You must provide the awscli command via entrypoint.
- Docker registry
  - Tag and push the docker image that you built to DockerHub. Don't forget to **docker login** first ;)
- Docker in docker.
  - Here we want you to know the docker inception capabilities, so please take a look at this page: <https://github.com/jpetazzo/dind>
  - After you have your first docker blow mind, you can go and test that images to play with docker in docker.
- Sharing the docker daemon
  - Now that you know the inception of docker we are going to learn, how to share your host daemon to create more dockers, instead of doing docker in docker. So the task here is to create a dockerfile, **from alpine** image base, and install the docker in it, but what we want is that when you execute the docker run command, you share the docker daemon with -v option, so after that you can start creating more containers from this containers.

- Writing your first docker compose file
  - Before to deep dive in the orchestrators tools, we are going to start here with the most basic, yes **docker-compose** , you will need to do the following.
    - Create a dockerfile that will have python installed on it (**from python** base image), and install the flask framework.
    - Create an redis image (**from redis** base image) that will serve as the database of the python app.
    - Create an nginx dockerfile (**from nginx** base image) that will be useful for serve all your static files.
    - Create an nginx dockerfile (**from nginx** base image) that will work as your proxy server.

With all of this setup, what you need to do is the following.

The application logic, can be whatever you like, make it simple :)

These are few requirements.

- The python app will need to talk with the redis database.
- The static content needs to be set on an nginx container.
- There should be a nginx container different from the above one, that will serve all the static files.

All of this configuration, needs to be automated with docker-compose files.

**If you are sure that you know all of the above things, speak with your manager, he will evaluate you in 20 minutes, so you can move forward to the following section.**

# Advanced docker exercises

## Before to start...

- Complete the next course  
<https://www.udemy.com/kubernetes-from-a-devops-kubernetes-guru>
- Talk with your manager and schedule a ECS deep tutorial with him .
- <https://medium.com/boltops/gentle-introduction-to-how-aws-ecs-works-with-example-tutorial-cea3d27ce63d>

Please only focus on seeing the videos, you will have more time to practice after finish them.

- Learning ECS
  - Basics ECS
    - Create an ECR repo
    - Push a nginx container to it
    - Create a ECS cluster using 2 private subnets in different AZ.
    - Create a task definition
    - Expose you nginx container through a public ALB
    - Once your application is exposed, create a service to launch many nginx containers into your cluster
  - Advanced ECS

Here we are going to put what did you do with docker compose on ECS , and we are going to build it with cloudformation

    - Create an ECS cluster
    - Create an autoscaling group for your ECS cluster.
    - Replace your nginx reverse proxy for an AWS ALB
    - Create a service for the python app (Desired count 2)
    - Create a service for the static files
    - Attach cloudwatch logs to all the services.
    - Use AWS elasticache redis
- Learning basics kubernetes
  - Basics Kubernetes (hands on lab)
    - Install minikube
    - Deploy your first application  
<https://github.com/dockersamples/example-voting-app>
    - Deploy the application the same application with EKS