Absolute Beginner’s Guide to Docker – Webinar

<https://blog.jetbrains.com/dotnet/2021/06/25/absolute-beginner-s-guide-to-docker-webinar-recording/>

<https://vishnuch.tech/docker-cheatsheet>

# Containers vs. Virtual Machines

## What is a VM?

* A piece of software that runs on a physical computer and that software pretends to be another computer.
* Pretends/emulates to be hardware

## What is a container?

* Not the same as a VM, but you can think of them as a VM.
* Typically faster and smaller than VMs.
* Don’t emulate HW, don’t emulate their own OS.
* Less computationally intensive.

# Containers vs. Images

* With a container (think VM), you have some files on disk that represent that VM. From those files, you can create a VM instance (or multiple instances).
* Same kind of thing with a Docker image. From that image, you can create multiple Docker containers.

# Linux Containers vs. Windows Containers

* Most of the time when searching on the internet about containers you will find stuff on Linux containers.

# Docker in Dev Examples

* docker run <image\_name>
  + Gateway to most of the common Docker functionality
  + Allows you to run docker containers
  + Is there a <image\_name> image on my local computer to spin up a container? If not, go check the image registry (DockerHub).
* docker ps
  + see running containers
  + -a: See all containers (even ones that are not running)
* docker rm <name>
  + Remove a container
* docker run –rm <name>
  + When the container stops, auto remove it
* docker stop <container>
  + Stop a running container
* docker exec -it <container\_id> <command>
  + exec runs a command inside an already running docker container
  + -it means interactive

# Running a Service in Docker

* docker run –rm -p <LOCAL\_MACHINE\_PORT>:<IMAGE\_PORT> <image\_name>
  + Map a port on your local machine to a port on the image image.
  + E.g., docker run –rm -p 8080:80 nginx
    - You can then curl <http://localhost:8080> and get a response from the running nginx docker image
* docker run –rm -d -p 15672:15672 rabbitmq:3-management
  + -d is run in detached mode (I.e., give you back control of the terminal window)
  + Note that often containers that run a service will have their own web server built in so that you can access the service from a browser.
    - E.g., in this case, <http://localhost:15672>
* docker run –rm -d -e POSTGRES\_HOST\_AUTH\_METHOD=trust postgres
  + -e is used for setting environment variables.
  + docker exec -it <container> \bin\bash
    - Think about this as sshing into the container
    - Then (inside the container) psql -U postgres
      * You can then fool around with postgres

# Docker Compose

* docker compose or docker-compose (both work, space is more recent)
* Lets you spin up a bunch of different containers (as services) all at the same time.
* Note on docker volumes and persisting data
  + <https://docs.docker.com/storage/volumes/>
  + <https://docs.docker.com/compose/compose-file/compose-file-v3/#volume-configuration-reference>
* docker-compose.yml file
  + <https://docs.docker.com/compose/>
* docker compose up -d
  + Start up based on docker-compose.yml file
  + -d flag is detached (i.e., give you back control of the terminal)