Usecase where docker is most useful

1. You are a developer in a team
2. Want to quickly deploy application to qa environment

Step 1: check if docker is installed

Graphical user interface, text, application

Description automatically generated

Docker run in28min/todo-rest-api-h2:1.0.0.RELEASE

A spring boot app is bring launched up!

1. But java is not installed ? how is it running ?
2. Docker – makes deploying applications a cake walk!

Docker behind the scenes

1. Docker run in28min/todo-rest-api-h2:1.0.0.RELEASE

Image is downloaded from hub.docker.com 🡪 docker registry (contains a lot of repos)

Hub.docker.com is a public registry 🡪 anyone can access

Hub.docker.com/in28min/todo-rest-api-h2 🡪 this is a repository

The repository hosts a multitude of tags 🡪 which are versions

What does image contain 🡪 all the things that the application needs to run

Image – a static template 🡪 a set of bytes

When the image is downloaded 🡪 and when it is running 🡪 then it is called a container

Docker run -p 5000:5000 in28min/todo-rest-api-h2:1.0.0.RELEASE

Hostport 🡪 first

Containerport 🡪 second

To run the application , you need to specify both hostport and containerport

You don’t want the application to be tied to the terminal

Docker run -p 5000:5000 -d in28min/todo-rest-api-h2:1.0.0.RELEASE

-d means detached

Docker container ls

See all the running containers

Docker run -p 5001:5000 -d in28min/todo-rest-api-h2:1.0.0.RELEASE

To run another container of the same image

Docker images

To show all the images present LOCALLY that have been pulled from docker hub

Docker container stop <container id>

To stop the container from ruynning

Diagram

Description automatically generated

Why is docker popular?

Chart, treemap chart

Description automatically generated

Chart

Description automatically generated with low confidence

Vms are heavyweight – need guest os and host os!

Docker dun need guest os

Docker tag in28min/todo-rest-api-h2:1.0.0.RELEASE in28min/todo-rest-api-h2:latest

Docker pull just pulls, but doesn’t run

Docker official images 🡪 curated set of docker repos that are stored on dockerhub

Docker image history 🡪 instructions run to create that image

Docker image inspect

Docker image remove

Docker run -p 5000:5000 -d in28min/todo-rest-api-h2:1.0.0.RELEASE

This command creates a container

Docker container pause <contained id>

Docker container inspect <container id>

Docker container prune 🡪 remove all stopped containers

Docker container ls -a

Will show containers of all statuses

Docker container stop 1b1 🡪 graceful shutdown

Docker container kill <id>

This is not graceful! It will not graceful

Docker run -p 5000:5000 -d –restart=always in28min/todo-rest-api-h2:0.0.1-SNAPSHOT

When u restart docker desktop, the container starts up automatically

Docker events 🡪 stuff that happens as and when something happens in each container

Docker stats 🡪 stats regarding running containers

When u run a docker contain, you can allocate cpu quota to it and also memory stuff

Distributed tracing with docker

How to debug problems

How to trace requests across microservices?

Enter distributed tracing

Diagram

Description automatically generated

Docker run -p 9411:9411 openzipkin/zipkin:2.23

Text

Description automatically generated

Diagram

Description automatically generated

Tracing server is now up!

Graphical user interface, text, application

Description automatically generated

Now, we need to connect all our microservices to zipkin

First thing we need to do is, add dependency to currency-exchange microservice

Graphical user interface, application

Description automatically generated