**Ports**

| **Application** | **Port** |
| --- | --- |
| Limits Service | 8080, 8081, ... |
| Spring Cloud Config Server | 8888 |
|  |  |
| Currency Exchange Service | 8000, 8001, 8002, .. |
| Currency Conversion Service | 8100, 8101, 8102, ... |
| Netflix Eureka Naming Server | 8761 |
| Netflix Zuul API Gateway Server | 8765 |
| Zipkin Distributed Tracing Server | 9411 |

Currency Exchange Service

http://localhost:8000/currency-exchange/from/USD/to/INR

Currency Conversion Service

http://localhost:8100/currency-conversion/from/USD/to/INR/quantity/10

http://localhost:8100/currency-conversion-feign/from/USD/to/INR/quantity/10

Eureka

http://localhost:8761/

API GATEWAY

http://localhost:8765/CURRENCY-EXCHANGE/currency-exchange/from/USD/to/INR

http://localhost:8765/CURRENCY-CONVERSION/currency-conversion/from/USD/to/INR/quantity/10

http://localhost:8765/CURRENCY-CONVERSION/currency-conversion-feign/from/USD/to/INR/quantity/10

http://localhost:8765/currency-exchange/currency-exchange/from/USD/to/INR

http://localhost:8765/currency-conversion/currency-conversion/from/USD/to/INR/quantity/10

http://localhost:8765/currency-conversion/currency-conversion-feign/from/USD/to/INR/quantity/10

http://localhost:8765/currency-exchange/from/USD/to/INR

http://localhost:8765/currency-conversion/from/USD/to/INR/quantity/10

http://localhost:8765/currency-conversion-feign/from/USD/to/INR/quantity/10

http://localhost:8765/currency-conversion-new/from/USD/to/INR/quantity/10

Test docker image

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

* To run a docker image in detached mode use -d
* To see log of container

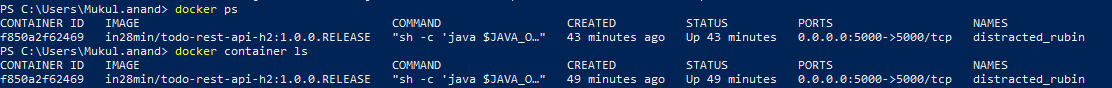
**docker logs containerid**

* To keep tailing the logs

**docker logs -f containerid**

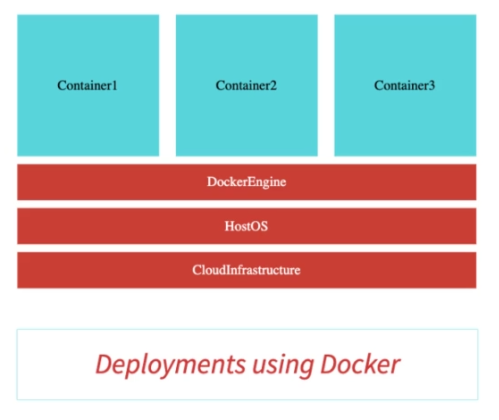
* To list all the containers running

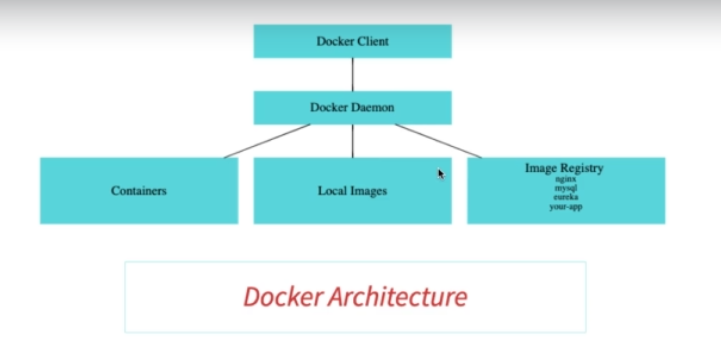
**docker container ls**

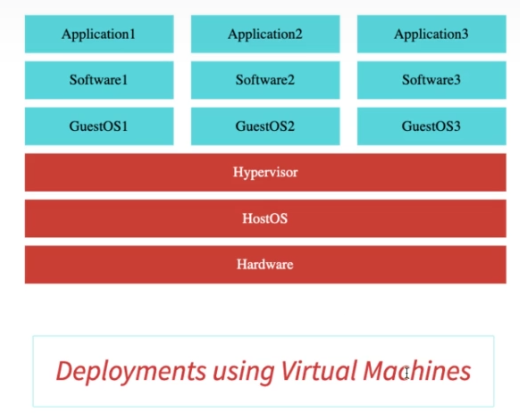


* To list all the containers

**docker container ls -a**





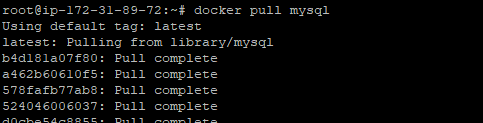


* To tag an image to any version or latest (latest represents to latest release but can point to any release)

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

* By default, docker pulls latest tag.

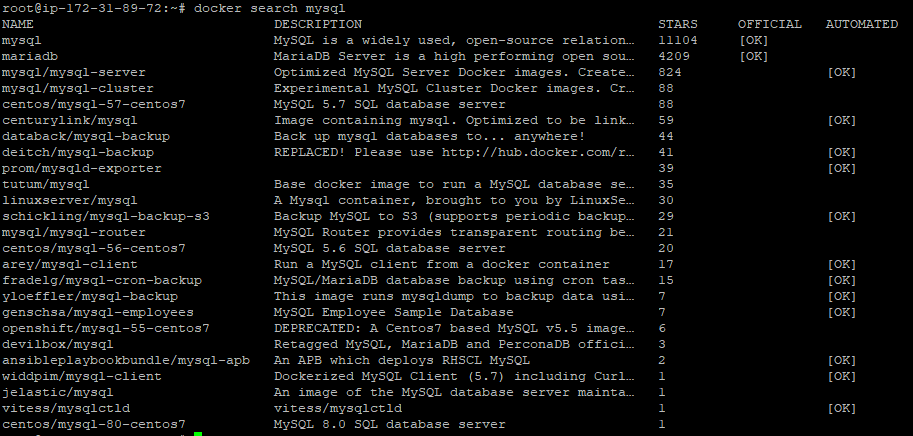
**docker pull mysql**



* To search any image

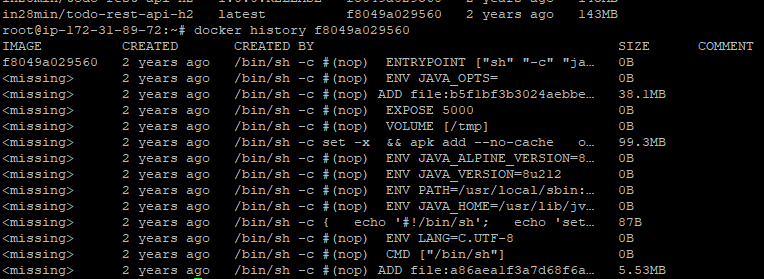
**docker search mysql**

It will show all related results –



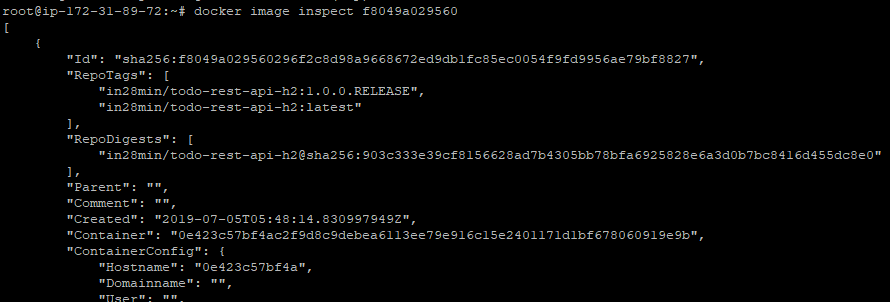
* To check steps involved in creating image we can use –

**docker history imageid**



* To check more details of image like OS, port exposed, tags, network –

docker image inspect imageid

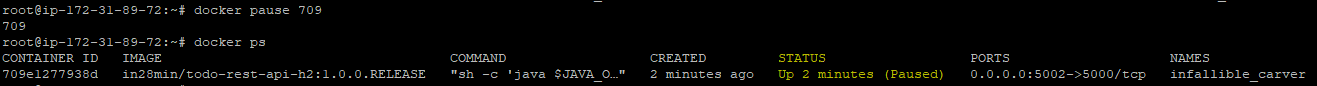


* To run a container, we can also use –

**docker container run -p 5002:5000 in28min/todo-rest-api-h2:1.0.0.RELEASE**

* We can pause a running container. (It will be running but not accesable)

**docker pause containerid**

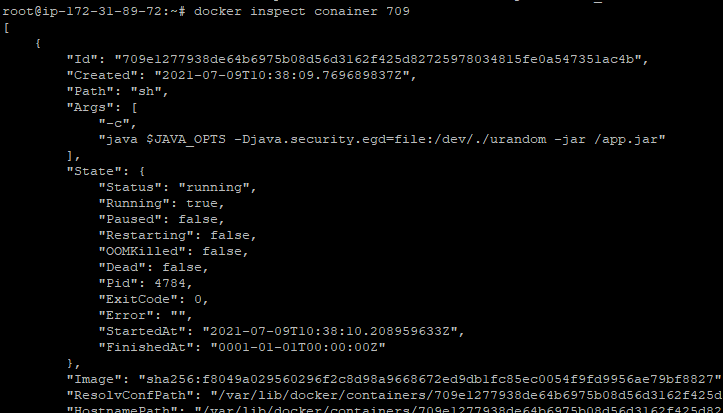


* We can also unpause the container

**docker unpause containerid**

* We can also inspect container

**docker container inspect containerid**



* To stop a container gracefully (SIGTERM)

**docker container stop containerid**

* To stop container forcefully (SIGKILL)

**docker container kill cointainerid**

* To start a container on docker start (default restart policy is no and another is always)

Useful in database kind of containers which should always running even on accidental docker stop and start.

**docker run -p 5000:5000 -d - -restart=always in28min/todo-rest-api-h2:1.0.0.RELEASE**

* To check events happening after command. Like docker stop

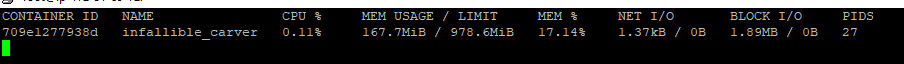
**docker events**

* To check top running process of a container

**docker top**

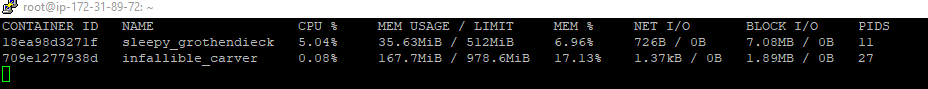
* To check memory and cpu usage

**docker stats**



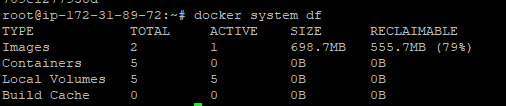
* To assign cpu and memory we can use –

**docker run -p 5003:5000 -d -m 512m - -cpu-quota 5000 in28min/todo-rest-api-h2:1.0.0.RELEASE**



* To check what all docker demon manages

**docker demon df**



**docker run -p 9411:9411 openzipkin/zipkin:2.23**

**In currency-exchange-service add –**

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-sleuth</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-sleuth-zipkin</artifactId>

</dependency>

Application.properties

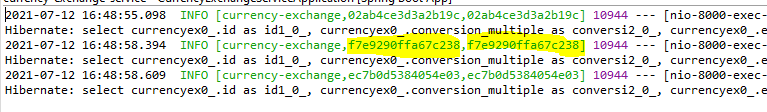
#How much logging trace you want. 1.0 is 100%. 0.1 is 10%. Based on requirement we add

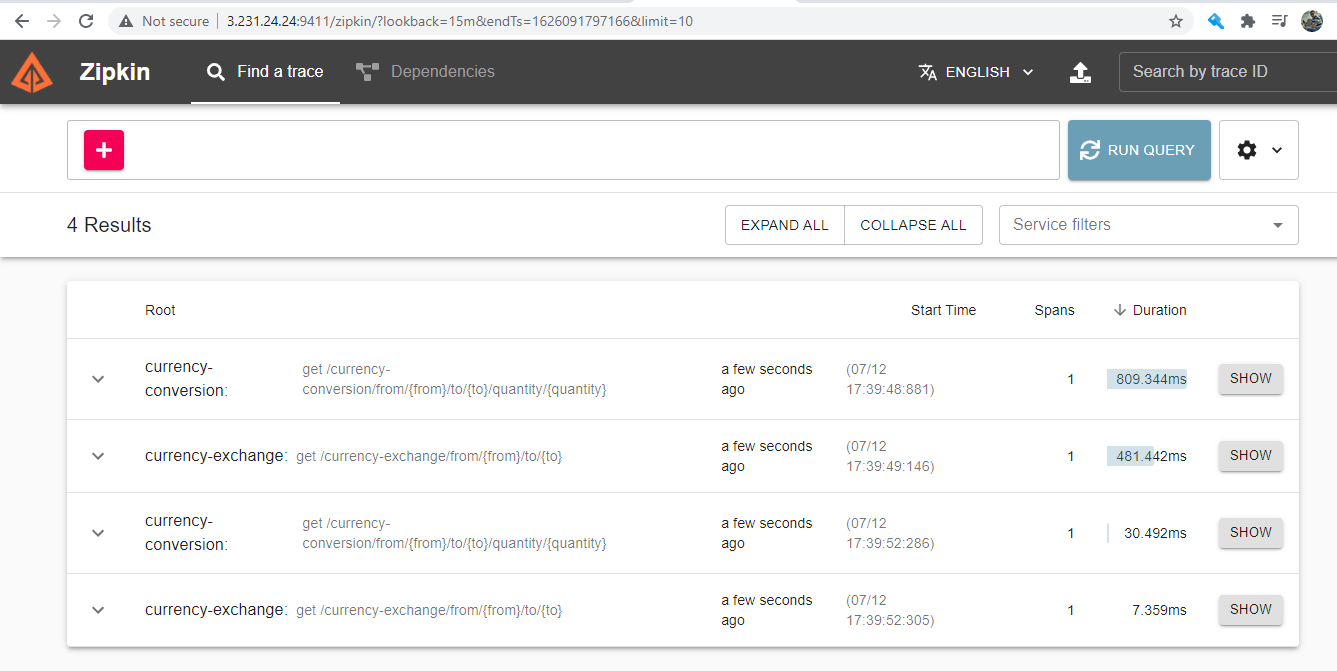
spring.sleuth.sampler.probability=1.0

#spring.zipkin.baseUrl=http://localhost:9411/

spring.zipkin.baseUrl=http://3.231.24.24:9411/

We can see id is getting assigned in logs –





How to add plugin to build image in POM.xml

Add in Pom.xml –

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<configuration>

<image>

<name>in28min/mmv2-${project.artifactId}:${project.version}</name>

</image>

<pullPolicy>IF\_NOT\_PRESENT</pullPolicy>

</configuration>

</plugin>

</plugins>

</build>

Go to Run As -> build.. -> Goals and enter below command –

spring-boot:build-image -DskipTests

then run