PETER-SERVICE

HOW CAN WHALE EAT A JAVA APP

AND DO NOT CHOKE

ABOUT ME

Aleksandr Fedorov

Sr. SE at PETER-SERVICE

More then 10 years with Java

2 years hugs with Docker

SPbPU PhD in progress..

PETER-SERVICE

PRESS START

PETER-SERVICE

ADVERTISEMENT TIME

AGILITY

ACCELERATE SOFTWARE DEVELOPMENT AND DEPLOYMENT BY 13X AND RESPOND INSTANTLY TO CUSTOMER NEEDS.

PORTABILITY

ELIMINATE THE “WORKS ON MY MACHINE” ONCE AND FOR ALL.
GAIN INDEPENDENCE ACROSS ALL ENVIRONMENTS.

PETER-SERVICE

LEVEL 1

Simple app

JAVA

DOCKER

LEVEL 1

PETER-SERVICE

HelloJavaMeetup.java

```
public class HelloJavaMeetup{
public static void
main(String[] args) {
System.out.println("Hello Java
Meetup!");
}
}
```

Dockerfile

```
FROM java:8
ADD HelloJavaMeetup.java .
RUN javac HelloJavaMeetup.java
CMD ["java", "HelloJavaMeetup"]
```

Docker commands

```
$ docker build -t java-app:demo .
$ docker images
$ docker run java-app:demo
```

Tips about Base image

From Arun Gupta (DockerCon17):

- Java base image - don't use java:8
- Prefer openjdk:8 or openjdk:9
- Debian or Alpine (2 times smaller!)
- Use jre for running jars, build by jdk

PETER-SERVICE

LEVEL 2

Integration

DOCKER

MAVEN/GRADLE

PETER-SERVICE

LEVEL 2

Maven or Gradle plugin

fabric8io/docker-maven-plugin

spotify/docker-maven-plugin

VS

bmuschko/gradle-docker-plugin

transmode/gradle-docker

...

PETER-SERVICE

Maven or Gradle plugin

bmuschko/gradle-docker-plugin

```
docker {
    javaApplication {
        baseImage = 'openjdk:latest'
        tag = 'java-app:gradle' }
}
task createContainer(type: DockerCreateContainer) {
    dependsOn dockerBuildImage
        targetImageId { dockerBuildImage.getImageId() }
}
task startContainer(type: DockerStartContainer) {
    dependsOn createContainer
    targetContainerId { createContainer.getContainerId() }
}
```

PETER-SERVICE

LEVEL 3

Multi-container app

Multi-container app

docker-compose.yml

```yaml
version: '3.3'
services:
  postgres:
      image: postgres:10.1
      ports:
          - 5432:5432
      restart: always
  app:
      image: app
      depends_on:
        - postgres
      ports:
          - 8080:8080
      restart: always
```

Docker compose commands

```
$ docker-compose run -d
$ docker-compose stop
```

PETER-SERVICE

BONUS LEVEL

Docker SWARM

Multi container app

Docker SWARM

Docker commands in swarm mode

```
$ docker-compose scale jm-app=3
$ docker stack deploy -c docker-compose.yml javameetup
```

PETER-SERVICE

FINAL LEVEL

JAVA

DOCKER

FINAL LEVEL

Memory management tips

- --memory doesn't work as expected
  - App will be killed by docker
  - No out of memory exception
- Always put with -Xmx
- --memory should be 2 times larger than -Xmx
- Jdk9 support cgroups memory limits

Memory leak simulation

https://github.com/valentinomiazzo/docker-jvm-memory-test/blob/master/README.md

```
docker run -it --memory=32m --memory-swap=32m --env ALLOC_HEAP_MB=1
--env MAX_HEAP_SIZE_MB=128 valentinomiazzo/jvm-memory-test

docker run -it --memory=64m --memory-swap=64m --env ALLOC_HEAP_MB=1
--env MAX_HEAP_SIZE_MB=32 valentinomiazzo/jvm-memory-test

docker inspect -f '{{json .State}}' $(docker ps -l -q)
```

Pros and Cons

-0.1 second libcontainer

+Less storage and memory

-Cgroups and namespaces

+Configuration as a code

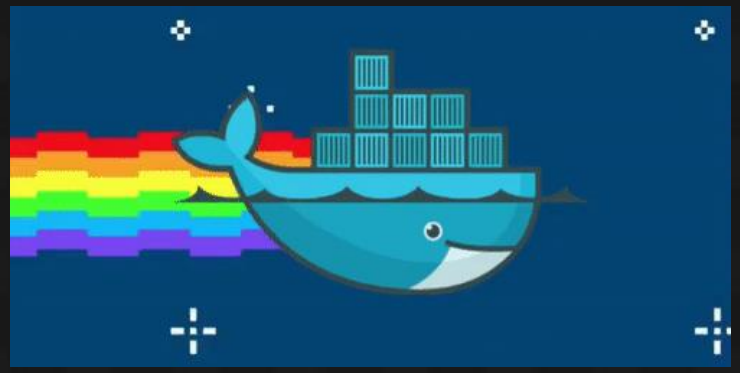
Java 9

The JVM will now consider cgroups memory limits if the following flags are specified:

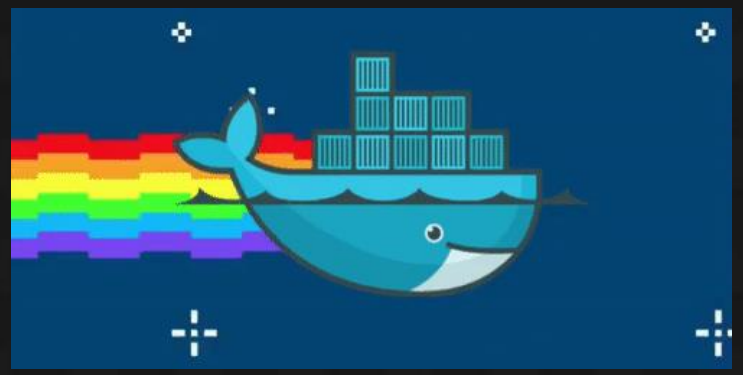
-XX:+UseCGroupMemoryLimitForHeap

-XX:+UnlockExperimentalVMOptions

GAME OVER



Q/A



Thank you!

LINKS

https://github.com/alexff91/java-meetup-2017

Level 1:

https://www.youtube.com/watch?v=yHLAaA4gPxw

https://www.tutorialkart.com/docker/docker-java-application-example/

Level 2:

https://github.com/fabric8io/docker-maven-plugin

https://github.com/bmuschko/gradle-docker-plugin

Level 3:

https://github.com/jirkapinkas/spring-boot-postgresql-docker-compose

https://codefresh.io/docker-tutorial/deploy-docker-compose-v3-swarm-mode-cluster/

Final Level:

https://hackernoon.com/crafting-perfect-java-docker-build-flow-740f71638d63

https://jaxenter.com/nobody-puts-java-container-139373.html

https://github.com/valentinomiazzo/docker-jvm-memory-test