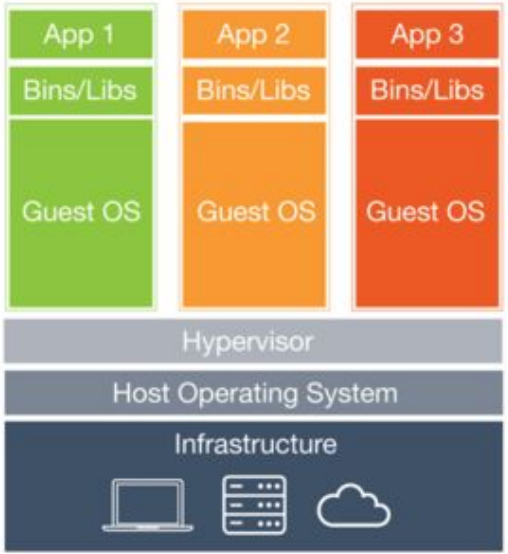
Java/JVM com Docker em produção: lições das trincheiras

Leonardo Zanivan
panga@apache.org

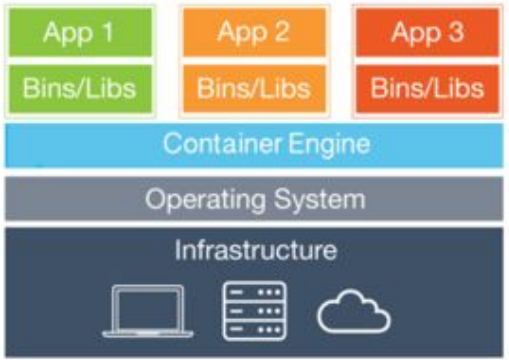
QCon
SÃO PAULO

Why ~~Docker~~ Container?

Review:



Hypervisor-based Virtualization



Container virtualization

QCon SÃO PAULO

Why ~~Docker~~ Container?

- Environments (dev, test, UAT, prod)
- Productivity (onboarding, develop, test)
- Single Responsibility Principle
- DevOps or Dev + Ops
- Economies of $cale

Use Cases

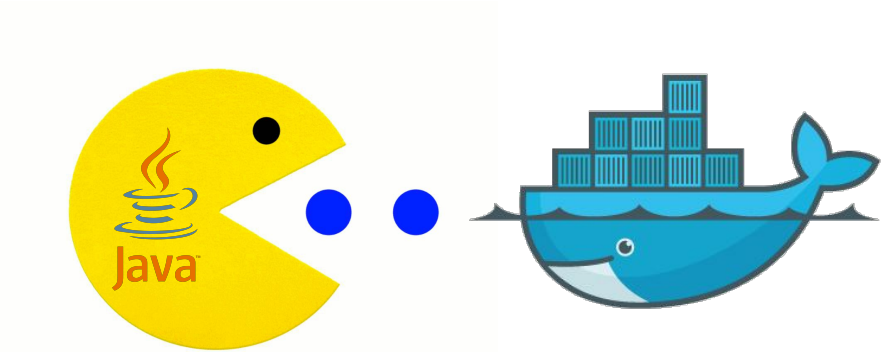
- Pokemon GO (1000+ nodes)
- "X" Messaging (1000+ containers)
- Uber Docker Host (~300 containers)

JVM + Containers (docker, rkt, runC)

- Memory
- CPU
- Disk I/O
- Network

JVM Memory on Container

- Common problems:
  - OOM Killer
  - OutOfMemory error
  - High memory usage



JVM Memory on Container

Cause #1: Java Max Heap Size not defined (*-Xmx*)

- JVM default MaxHeapSize = Total host memory / 4
- JVM isn't aware of *cgroups!* *(JDK 9 has an experimental flag)*

Example:

```
total host memory     = 32GB
max container memory  = 1GB
default heap size     = 8GB
```

QCon SÃO PAULO

JVM Memory on Container

Cause #2: Container Memory < Java Memory (Heap+Stack)

- Java max heap isn't the max amount of memory used
- Use a 0.7 factor of Java Max Heap to Container

Example:

```
max container memory = 1GB
wrong max heap size  = 1GB
ok max heap size     = 700MB
```

QCon SÃO PAULO

JVM Memory on Container

Cause #3: No SWAP partition

- Your local machine has SWAP, but production not!
- Default container SWAP limit on Docker is 2*memory

Example:

```
max container memory = 1GB
max container swap   = 2GB
max jvm heap size    = 2GB
```

QCon SÃO PAULO

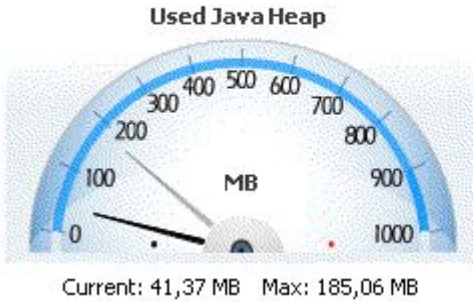
JVM Memory on Container

Cause #4: Default Garbage Collector

- Always specify a Garbage Collector (JDK < 9)
- Default GC doesn't scale, is slow and consume more RAM

Solution:

CMS = *-XX:+UseConcMarkSweepGC*
G1 = *-XX:+UseG1GC*



JVM CPU on Container

- *Problem:* Slow GC performance, bad lambda parallelism
- *Cause:* JVM isn't aware of *cgroups!*

Example:

```
total host cores     = 8
max container cores = 1
max jvm cores        = 8
```

JVM CPU on Container

- *Solution:* Set appropriate JVM properties

```
-XX:ParallelGCThreads=<max_container_cores>
-XX:ConcGCThreads=...
-Djava.util.concurrent.ForkJoinPool.common.parallelism=...
```

JVM Disk I/O on Container

- *Problem:* Slow WRITE performance
- *Cause:* Container is using graph driver
- *Solution*: Create a named volume or mount from host

```
docker volume create mysql-data
docker run -v mysql-data:/var/lib/mysql
```

JVM Disk I/O on Container

- *Problem:* Slow SecureRandom entropy calculation
- *Cause:* Container doesn't have enough events
- *Solution*: Set security JVM property to async

-Djava.security.egd=file:/dev/urandom

QCon SÃO PAULO

JVM Network on Container

- *Problem:* Bad DNS resolution on Alpine based images
- *Cause:* Alpine images doesn't use glibc
- *Solution*: Don't use Alpine images when using DNS reverse lookups or Domain Search

Example:

```
docker run --dns-search=service.consul
$ ping myservice
$ ping: cannot resolve myservice:
        Unknown host
```

QCon SÃO PAULO

IDE support for Docker

- NetBeans (8.2+)
- IntelliJ
- Eclipse

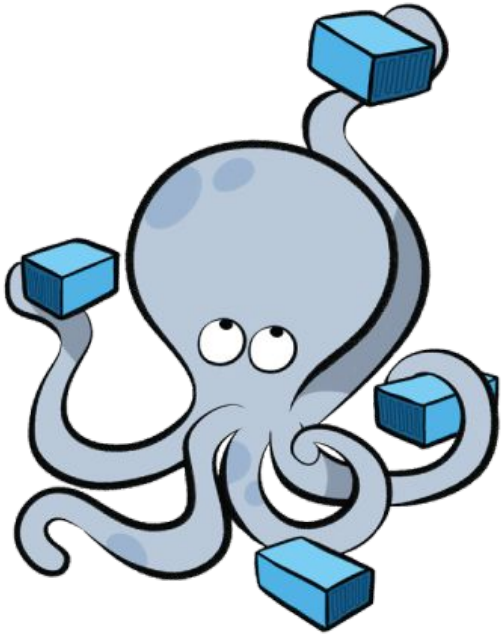
NetBeans

Tooling support for Docker

- Build lifecycle
  - Maven Plugin (docker-maven-plugin)
  - Gradle (gradle-docker-plugin)
- Tests
  - JUnit (docker-compose-rule)
  - Arquillian Cube

Container Schedulers

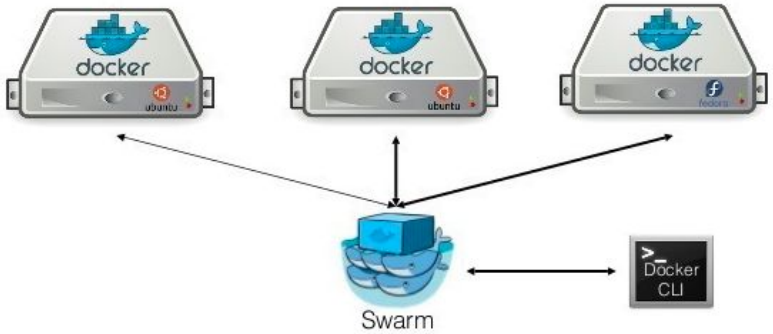
- Docker Swarm
- Kubernetes
- Mesos/Marathon
- AWS ECS
- etc.



Introducing Swarm + docker compose v3

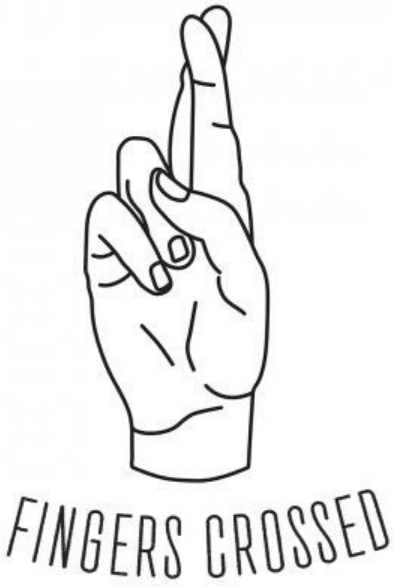
- Swarm is ready to use in Docker 1.13+
- Compose v3 support secrets & deploy options

```
secrets:
 -  mypassword:
deploy:
 -  replicas
 -  resources limits
 -  update config
 -  placement
```



QCon SÃO PAULO

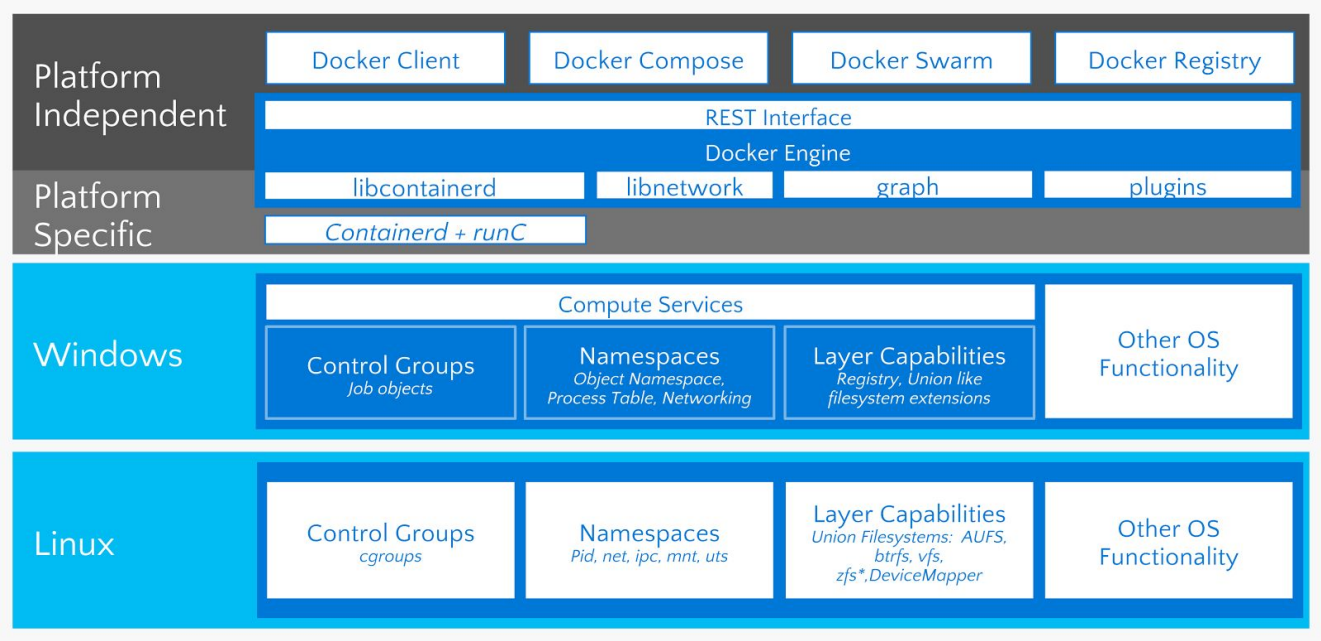
Demo time!



Extra Container challenges

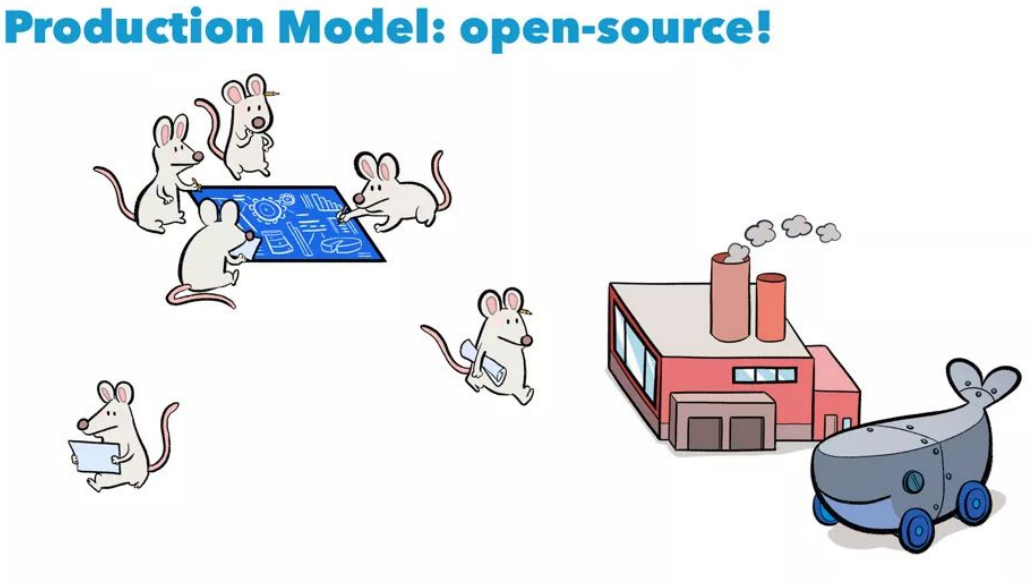
- Multi-host Networking
- Transparent Proxy
- Service Discovery
- Monitoring & Logs

Docker Architectural View



Moby Project

`github.com/`**`docker/docker`** `=>` `github.com/`**`moby/moby`**



Questions?

panga@apache.org
github.com/panga/qcon2017

QCon
SÃO PAULO