Docker for Java Developers
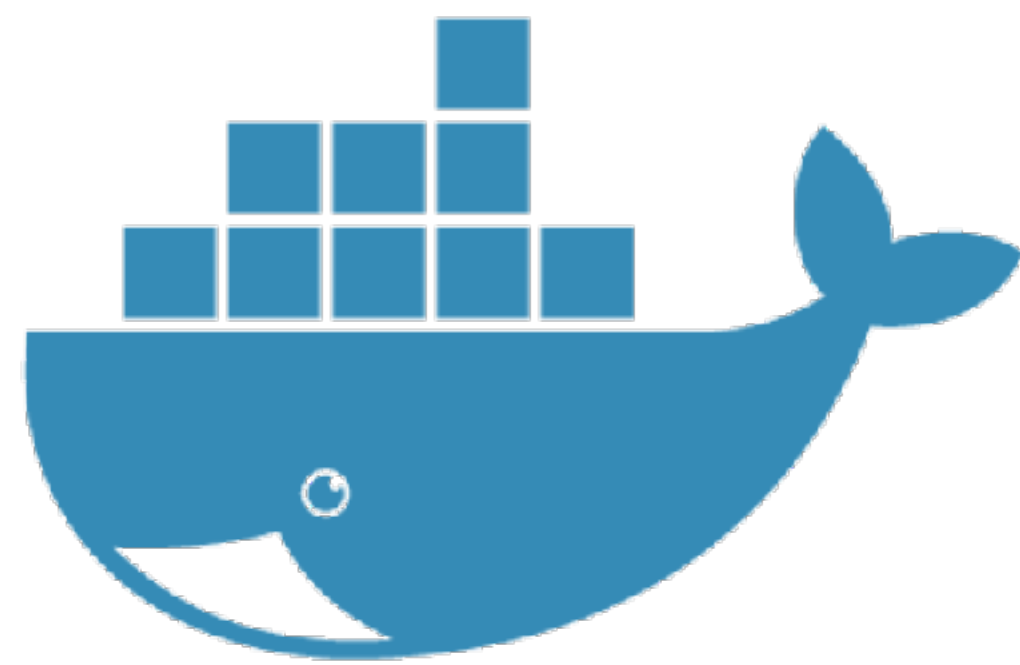
Fabiane Nardon, @fabianenardon
Arun Gupta, @arungupta

Java Champion

Duke's Choice Award Winner (2 years)

SouJava Founder

Data Scientist / Big Data expert

Docker Captain

Java Champion

JavaOne Rock Star (4 years)

NetBeans Dream Team

Silicon Valley JUG Leader

Author
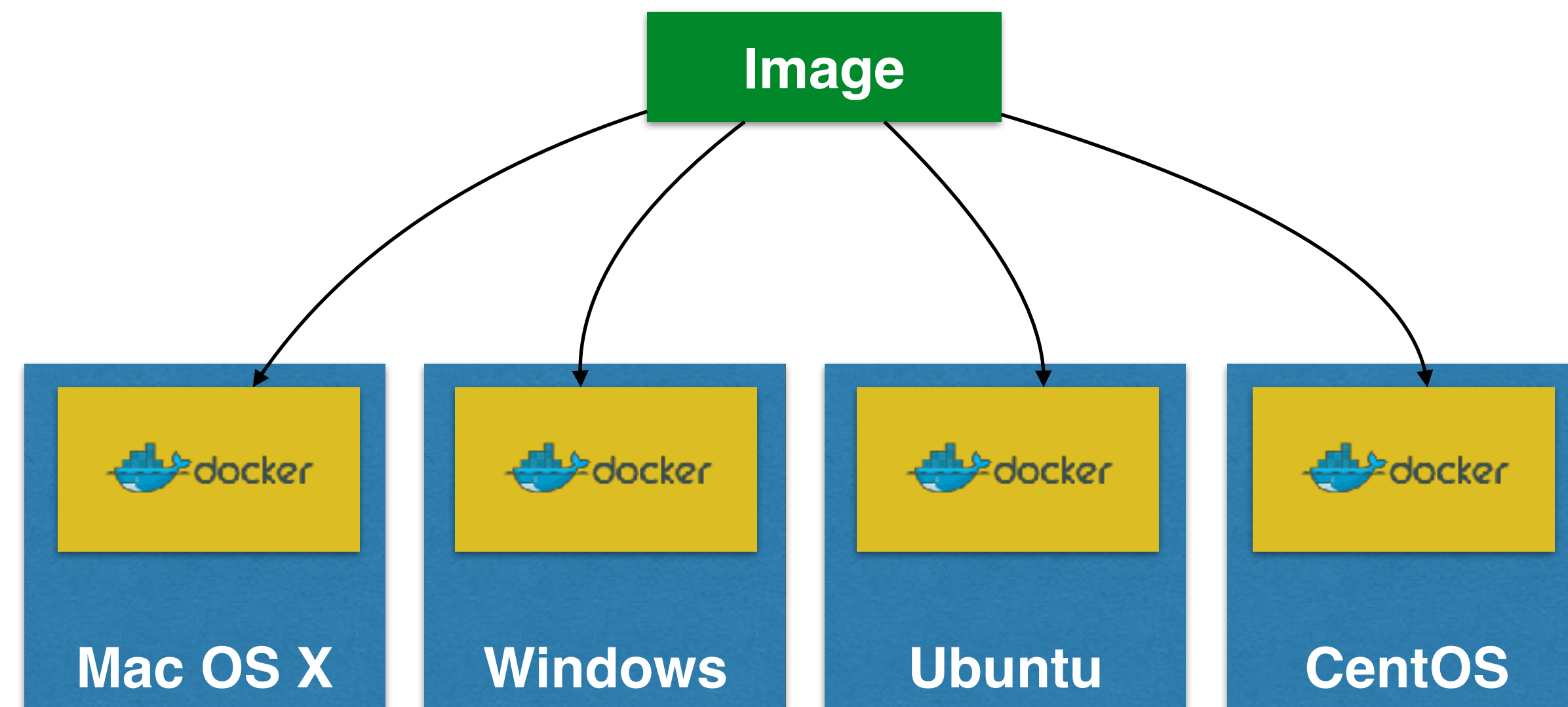
Runner

Lifelong learner

What we plan to cover?

- Java Base Image

- Package Java application with Maven and Gradle

- Multi-container application on single/multiple host(s)

- Scaling apps on AWS or Azure

- Memory management

- Debugging Java applications

- Monitor Java application

- Deployment pipeline for Java applications with Docker

Java Base Image
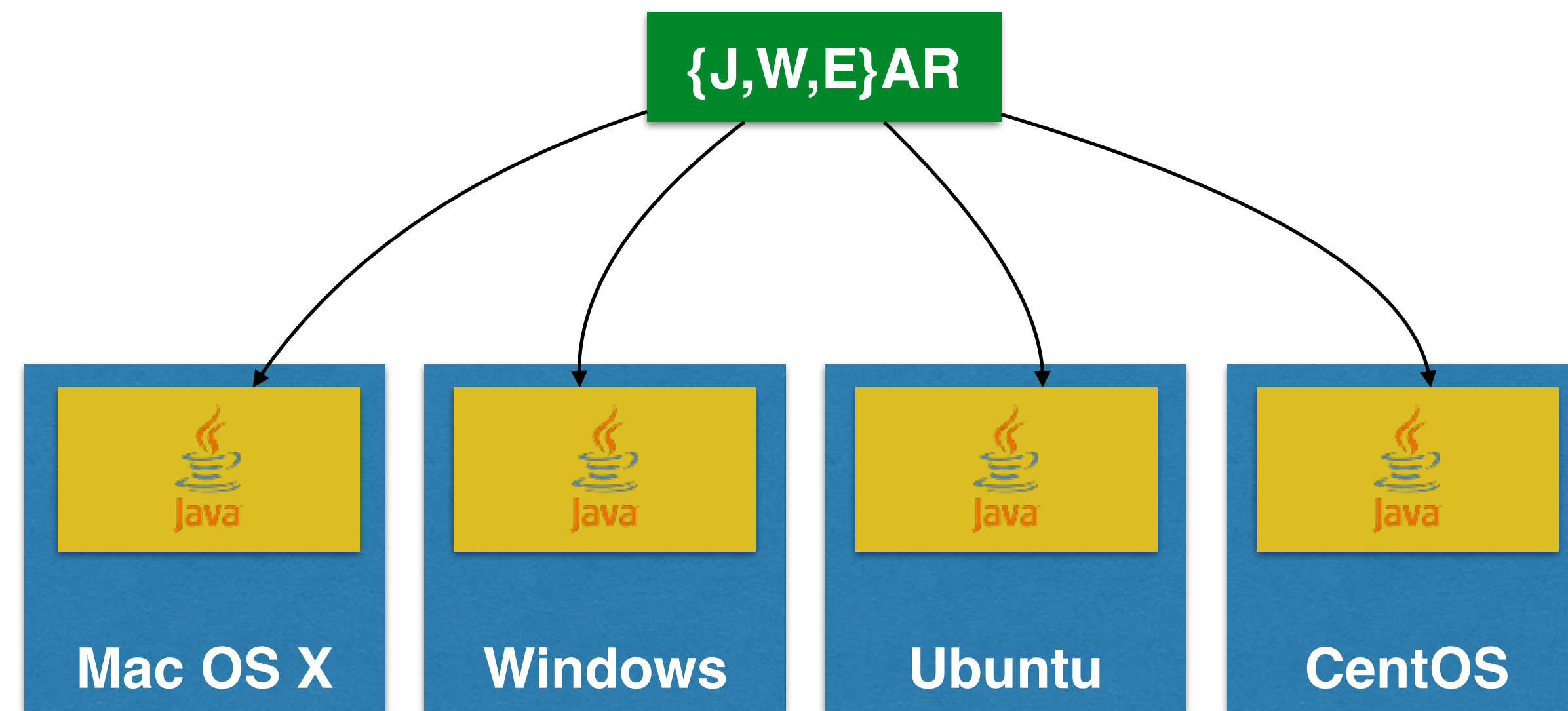
WORA = Write Once Run Anywhere

PODA = Package Once Deploy Anywhere

Java base image #1

Java base image #2

	Debian	Alpine
jdk	244MB	71MB
jre	124MB	56MB

Q openjdk

OFFICIAL REPOSITORY

openjdk ☆

Last pushed: 9 days ago

Repo Info   Tags

Short Description

OpenJDK is an open-source implementation of the Java Platform, Standard Edition

Full Description

Supported tags and respective `Dockerfile` links

- `6b38-jdk`, `6b38`, `6-jdk`, `6` (*6-jdk/Dockerfile*)
- `6b38-jre`, `6-jre` (*6-jre/Dockerfile*)
- `7u121-jdk`, `7u121`, `7-jdk`, `7` (*7-jdk/Dockerfile*)
- `7u121-jdk-alpine`, `7u121-alpine`, `7-jdk-alpine`, `7-alpine` (*7-jdk/alpine/Dockerfile*)
- `7u121-jre`, `7-jre` (*7-jre/Dockerfile*)
- `7u121-jre-alpine`, `7-jre-alpine` (*7-jre/alpine/Dockerfile*)
- `8u121-jdk`, `8u121`, `8-jdk`, `8`, `jdk`, `latest` (*8-jdk/Dockerfile*)
- `8u121-jdk-alpine`, `8u121-alpine`, `8-jdk-alpine`, `8-alpine`, `jdk-alpine`, `alpine` (*8-jdk/alpine/Dockerfile*)
- `8u121-jdk-windowsservercore`, `8u121-windowsservercore`, `8-jdk-windowsservercore`, `8-windowsservercore`, `jdk-windowsservercore`, `windowsservercore` (*8-*

```
38        cd /tmp 🔲 unzip /tmp/jce_policy-${JAVA_VERSION_MAJOR}.zip 🔲 \
39          cp -v /tmp/UnlimitedJCEPolicyJDK8/*.jar /opt/jdk/jre/lib/security; \
40        fi 🔲 \
41        sed -i s/#networkaddress.cache.ttl=-1/networkaddress.cache.ttl=10/ $JAVA_HOME/jre/lib/security/java.security 🔲 \
42        apk del curl glibc-i18n 🔲 \
43        rm -rf /opt/jdk/*src.zip \
44              /opt/jdk/lib/missioncontrol \
45              /opt/jdk/lib/visualvm \
46              /opt/jdk/lib/*javafx* \
47              /opt/jdk/jre/plugin \
48              /opt/jdk/jre/bin/javaws \
49              /opt/jdk/jre/bin/jjs \
50              /opt/jdk/jre/bin/orbd \
51              /opt/jdk/jre/bin/pack200 \
52              /opt/jdk/jre/bin/policytool \
53              /opt/jdk/jre/bin/rmid \
54              /opt/jdk/jre/bin/rmiregistry \
55              /opt/jdk/jre/bin/servertool \
56              /opt/jdk/jre/bin/tnameserv \
57              /opt/jdk/jre/bin/unpack200 \
58              /opt/jdk/jre/lib/javaws.jar \
59              /opt/jdk/jre/lib/deploy* \
60              /opt/jdk/jre/lib/desktop \
61              /opt/jdk/jre/lib/*javafx* \
62              /opt/jdk/jre/lib/*jfx* \
63              /opt/jdk/jre/lib/amd64/libdecora_sse.so \
64              /opt/jdk/jre/lib/amd64/libprism_*.so \
65              /opt/jdk/jre/lib/amd64/libfxplugins.so \
66              /opt/jdk/jre/lib/amd64/libglass.so \
67              /opt/jdk/jre/lib/amd64/libgstreamer-lite.so \
68              /opt/jdk/jre/lib/amd64/libjavafx*.so \
69              /opt/jdk/jre/lib/amd64/libjfx*.so \
70              /opt/jdk/jre/lib/ext/jfxrt.jar \
71              /opt/jdk/jre/lib/ext/nashorn.jar \
72              /opt/jdk/jre/lib/oblique-fonts \
73              /opt/jdk/jre/lib/plugin.jar \
74              /tmp/* /var/cache/apk/* 🔲 \
75        echo 'hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4' >> /etc/nsswitch.conf
76
77  # EOF
```

Java base image #3

https://hub.docker.com/_/openjdk/

Q openjdk

OFFICIAL REPOSITORY

openjdk ☆

Last pushed: 9 days ago

Repo Info   Tags

Short Description

OpenJDK is an open-source implementation of the Java Platform, Standard Edition

Full Description

Supported tags and respective `Dockerfile` links

- `6b38-jdk`, `6b38`, `6-jdk`, `6` (*6-jdk/Dockerfile*)
- `6b38-jre`, `6-jre` (*6-jre/Dockerfile*)
- `7u121-jdk`, `7u121`, `7-jdk`, `7` (*7-jdk/Dockerfile*)
- `7u121-jdk-alpine`, `7u121-alpine`, `7-jdk-alpine`, `7-alpine` (*7-jdk/alpine/Dockerfile*)
- `7u121-jre`, `7-jre` (*7-jre/Dockerfile*)
- `7u121-jre-alpine`, `7-jre-alpine` (*7-jre/alpine/Dockerfile*)
- `8u121-jdk`, `8u121`, `8-jdk`, `8`, `jdk`, `latest` (*8-jdk/Dockerfile*)
- `8u121-jdk-alpine`, `8u121-alpine`, `8-jdk-alpine`, `8-alpine`, `jdk-alpine`, `alpine` (*8-jdk/alpine/Dockerfile*)
- `8u121-jdk-windowsservercore`, `8u121-windowsservercore`, `8-jdk-windowsservercore`, `8-windowsservercore`, `jdk-windowsservercore`, `windowsservercore` (*8-*

Java base image #4

https://hub.docker.com/_/openjdk/

Q openjdk

OFFICIAL REPOSITORY

openjdk ☆

Last pushed: 9 days ago

Repo Info   Tags

Short Description

OpenJDK is an open-source implementation of the Java Platform, Standard Edition

Full Description

Supported tags and respective `Dockerfile` links

- `6b38-jdk`, `6b38`, `6-jdk`, `6` (*6-jdk/Dockerfile*)
- `6b38-jre`, `6-jre` (*6-jre/Dockerfile*)
- `7u121-jdk`, `7u121`, `7-jdk`, `7` (*7-jdk/Dockerfile*)
- `7u121-jdk-alpine`, `7u121-alpine`, `7-jdk-alpine`, `7-alpine` (*7-jdk/alpine/Dockerfile*)
- `7u121-jre`, `7-jre` (*7-jre/Dockerfile*)
- `7u121-jre-alpine`, `7-jre-alpine` (*7-jre/alpine/Dockerfile*)
- `8u121-jdk`, `8u121`, `8-jdk`, `8`, `jdk`, `latest` (*8-jdk/Dockerfile*)
- `8u121-jdk-alpine`, `8u121-alpine`, `8-jdk-alpine`, `8-alpine`, `jdk-alpine`, `alpine` (*8-jdk/alpine/Dockerfile*)
- `8u121-jdk-windowsservercore`, `8u121-windowsservercore`, `8-jdk-windowsservercore`, `8-windowsservercore`, `jdk-windowsservercore`, `windowsservercore` (*8-*

Java base image #5

openjdk	244MB	Debian
zulu-openjdk	161MB	Ubuntu

Q zulu

PUBLIC | AUTOMATED BUILD

azul/zulu-openjdk ☆

Last pushed: 2 months ago

Repo Info   Tags   Dockerfile   Build Details

Short Description

Zulu is a fully tested, compatibility verified, and trusted binary distribution of the OpenJDK.

Full Description

What is Zulu?

Zulu is a widely available binary distribution of OpenJDK. Zulu distributions are fully tested a
verified builds of the latest versions of the OpenJDK 8, 7, and 6 platforms. Zulu is available f
Linux, Windows, and MacOS platforms, with commercial support available upon request.

Zulu is built, tested, supported and made available by Azul Systems.

www.azul.com/zulu

Package Java Application using Maven or Gradle

First Java Application

```
FROM openjdk:jdk-alpine

CMD java -version
```

First Java Web Application

```
FROM jboss/wildfly:10.1.0.Final

COPY target/webapp.war /opt/jboss/wildfly/
standalone/deployments/webapp.war
```

Maven Plugin

- Plugin

  ```
  <groupId>io.fabric8</groupId>
  <artifactId>docker-maven-plugin</artifactId>
  <version>0.20.1</version>
  ```

- Goals: `docker:X`, X= `stop`, `build`, `push`, …

Maven - Configuration

```xml
63          <plugin>
64              <groupId>io.fabric8</groupId>
65              <artifactId>docker-maven-plugin</artifactId>
66              <version>0.20.1</version>
67              <configuration>
68                  <images>
69                      <image>
70                          <name>hellojava</name>
71                          <build>
72                              <from>openjdk:latest</from>
73                              <assembly>
74                                  <descriptorRef>artifact</descriptorRef>
75                              </assembly>
76                              <cmd>java -jar maven/${project.name}-${project.version}.jar</cmd>
77                          </build>
78                          <run>
79                              <wait>
80                                  <log>Hello World!</log>
81                              </wait>
82                          </run>
83                      </image>
84                  </images>
85              </configuration>
```

17

https://github.com/arun-gupta/docker-java-sample

Maven - Execution

```xml
 86                        <executions>
 87                            <execution>
 88                                <id>docker:build</id>
 89                                <phase>package</phase>
 90                                <goals>
 91                                    <goal>build</goal>
 92                                </goals>
 93                            </execution>
 94                            <execution>
 95                                <id>docker:start</id>
 96                                <phase>install</phase>
 97                                <goals>
 98                                    <goal>run</goal>
 99                                    <goal>logs</goal>
100                                </goals>
101                            </execution>
102                        </executions>
103                    </plugin>
104                </plugins>
```

Gradle Plugin

- Plugin: `com.bmuschko:gradle-docker-plugin:3.0.6`
- General purpose Docker Remote API
  - `DockerXImage`, X = `Build`, `Push`, `Remove`, …
  - `DockerXContainer`, X = `Create`, Start, `Stop`, `Kill`, …
- Opinionated Java application plugin
  - Extension properties: `baseImage`, `tag`, `port`, …

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

Gradle - Configuration

```
1    buildscript {
2        repositories {
3            jcenter()
4        }
5
6        dependencies {
7            classpath 'com.bmuschko:gradle-docker-plugin:3.0.6'
8        }
9    }
10
11   apply plugin: 'java'
12   apply plugin: 'application'
13   apply plugin: 'com.bmuschko.docker-java-application'
14
15   import com.bmuschko.gradle.docker.tasks.container.*
16   import com.bmuschko.gradle.docker.tasks.image.*
17
18   sourceCompatibility = 1.8
19   targetCompatibility = 1.8
```

Gradle - Execution

```
30    docker {
31        javaApplication {
32            baseImage = 'openjdk:latest'
33            tag = 'hellojava'
34        }
35    }
36
37    task createContainer(type: DockerCreateContainer) {
38        dependsOn dockerBuildImage
39        targetImageId { dockerBuildImage.getImageId() }
40    }
41
42    task startContainer(type: DockerStartContainer) {
43        dependsOn createContainer
44        targetContainerId { createContainer.getContainerId() }
45    }
```

Multi-Container Application
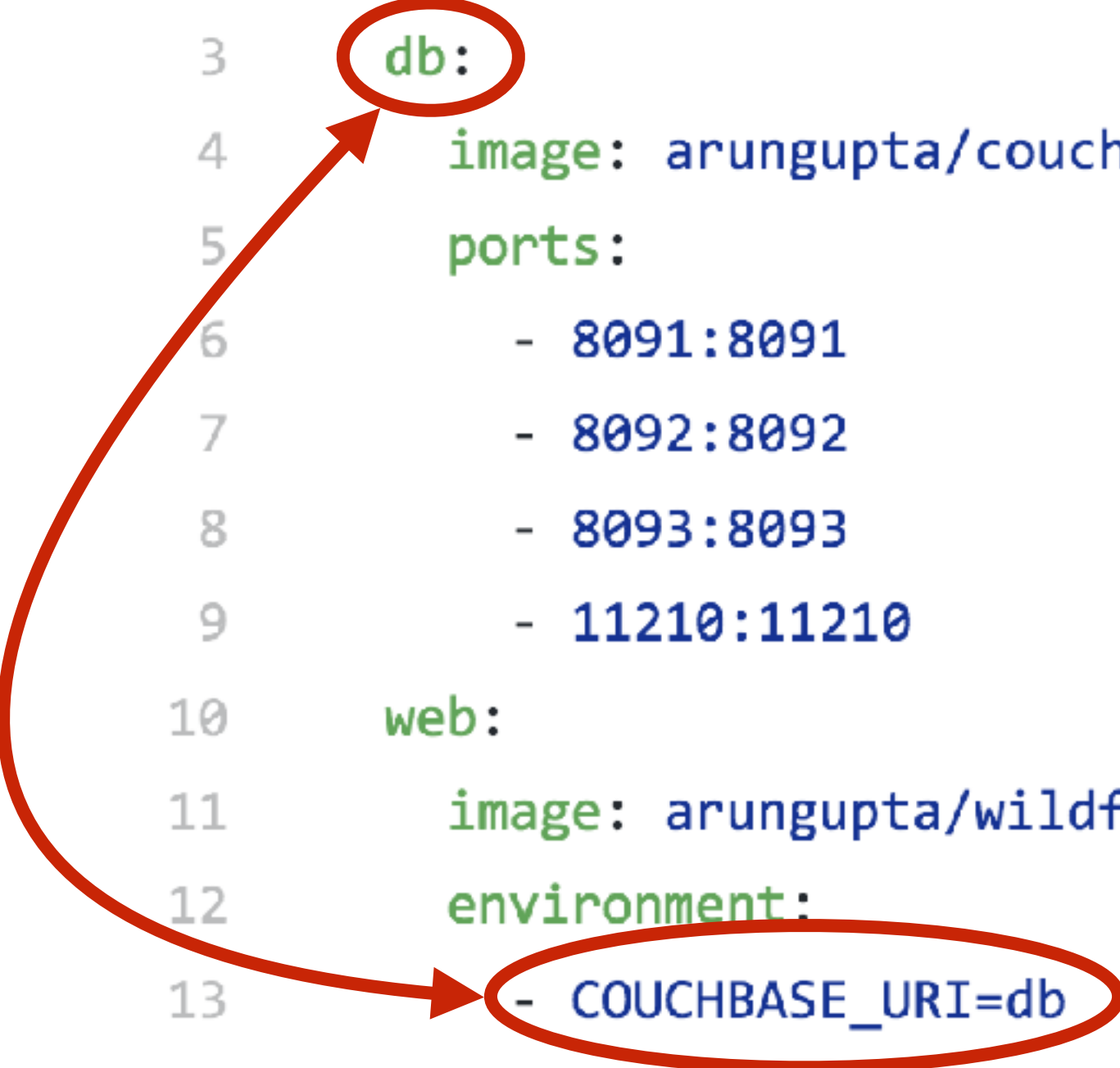
Docker Compose

- Define and run multi-container applications
- Configuration defined in one or more files
  - `docker-compose.yml` (default)
  - `docker-compose.override.yml` (default)
  - Multiple files specified using `-f`
- Single command to manage all services
- Great for dev, staging, and CI

Multi-container on single host

```
1    version: "3"
2    services:
3      db:
4        image: arungupta/couchbase:travel
5        ports:
6          - 8091:8091
7          - 8092:8092
8          - 8093:8093
9          - 11210:11210
10     web:
11       image: arungupta/wildfly-couchbase-javaee:travel
12       environment:
13         - COUCHBASE_URI=db
14       ports:
15         - 8080:8080
16         - 9990:9990
```

`docker-compose up`

Multiple Files - Image and Ports

docker-compose.db.yml

```
version: '3'
services:
  web:

  ports:
    - 80:8080
  db:
  image: couchbase:prod
  ports:
    - 8091:8091
```

Run

```
docker-compose \
-f docker-compose.yml \
-f docker-compose.db.yml \
up -d
```

Services

```
docker-compose \
-f docker-compose.yml \
-f docker-compose.db.yml \
ps
```

Shutdown

```
docker-compose \
-f docker-compose.yml \
-f docker-compose.db.yml \
down
```

Multi-host using Swarm-mode

Swarm Mode

- Natively managing a cluster of Docker Engines called a Swarm
- Docker CLI to create a swarm, deploy apps, and manage swarm
    - Optional feature, need to be explicitly enabled
- No Single Point of Failure (SPOF)
- Declarative state model
- Self-organizing, self-healing
- Service discovery, load balancing and scaling
- Rolling updates

Swarm Mode: Replicated Service
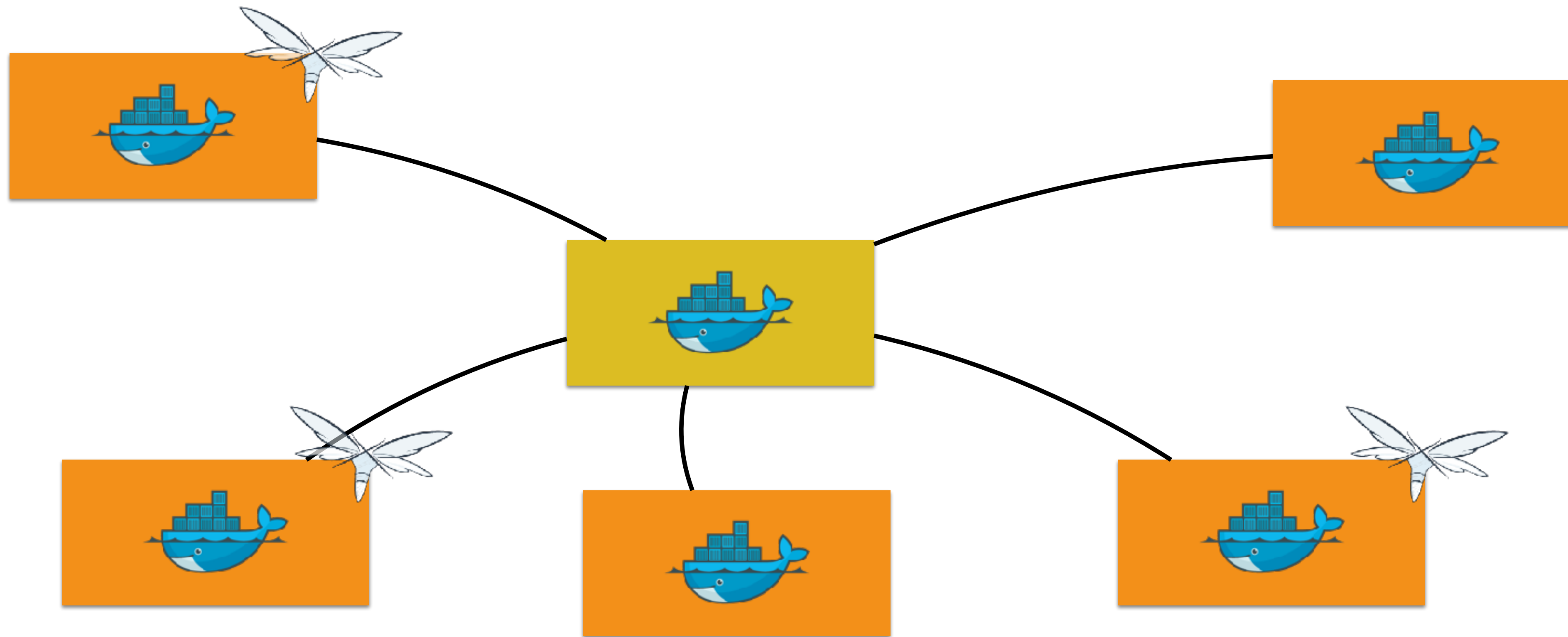

```
docker service create --replicas 3 --name web jboss/wildfly
```

Multi-container on multiple hosts

```yaml
1   version: "3"
2   services:
3     db:
4       image: arungupta/couchbase:travel
5       ports:
6         - 8091:8091
7         - 8092:8092
8         - 8093:8093
9         - 11210:11210
10    web:
11      image: arungupta/wildfly-couchbase-javaee:travel
12      environment:
13        - COUCHBASE_URI=db
14      ports:
15        - 8080:8080
16        - 9990:9990
```

`docker stack deploy --compose-file=docker-compose.yml webapp`

Swarm Mode: Routing Mesh
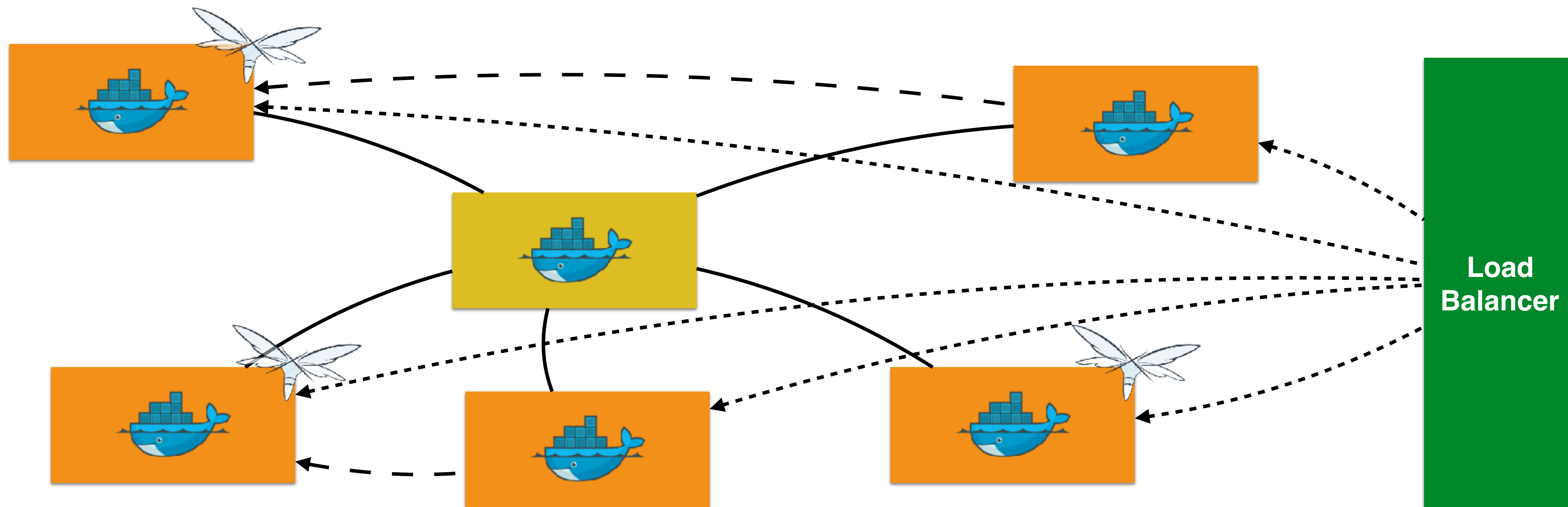

```
docker service create --replicas 3 --name web -p 8080:8080 jboss/
wildfly
```

Swarm Mode: Scale
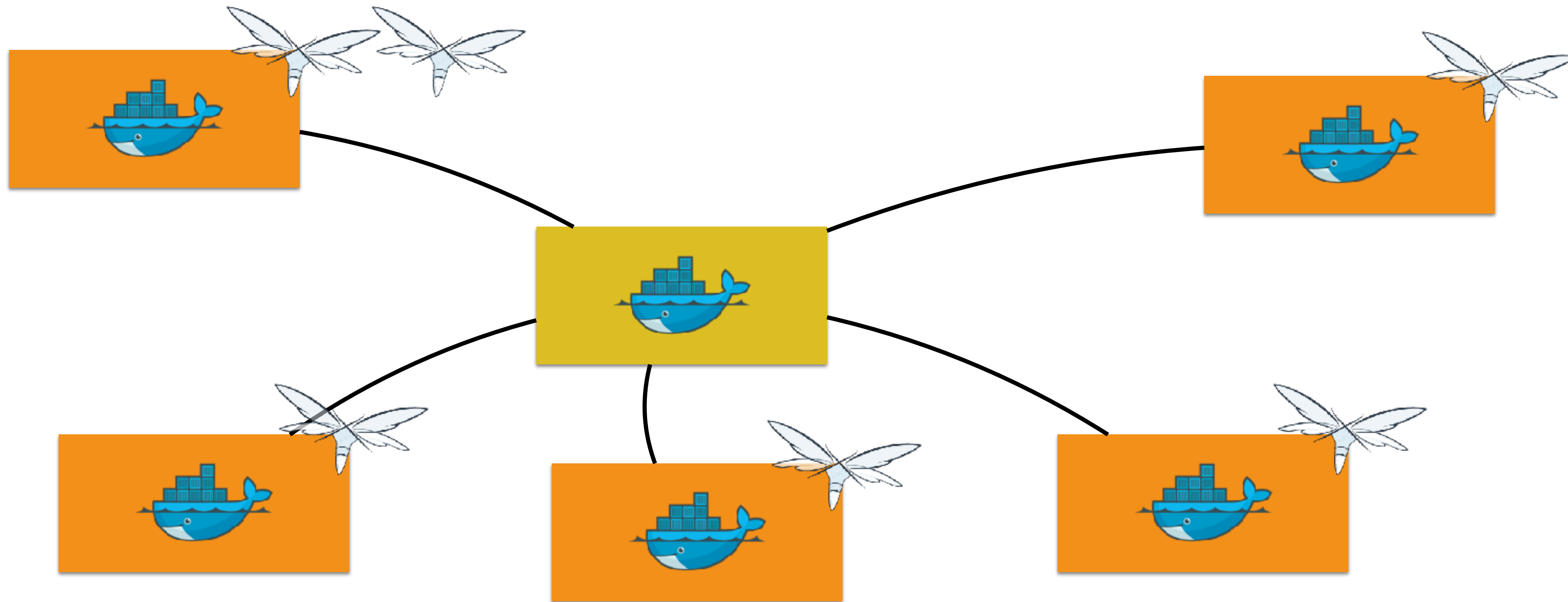
`docker service scale web=6`

Swarm Mode: Rolling Updates
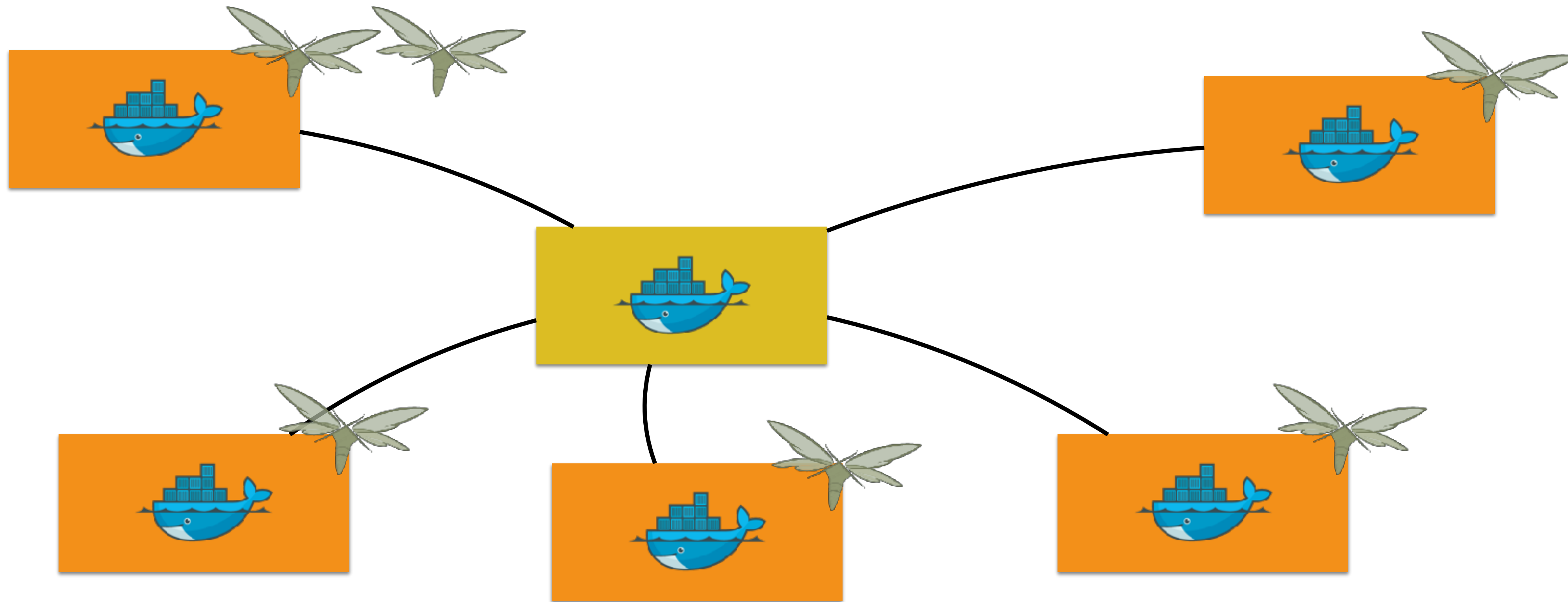

```
docker service update web --image wildfly:2 --update-parallelism
2 --update-delay 10s
```

Scaling Apps on AWS or Azure

Docker Community Edition
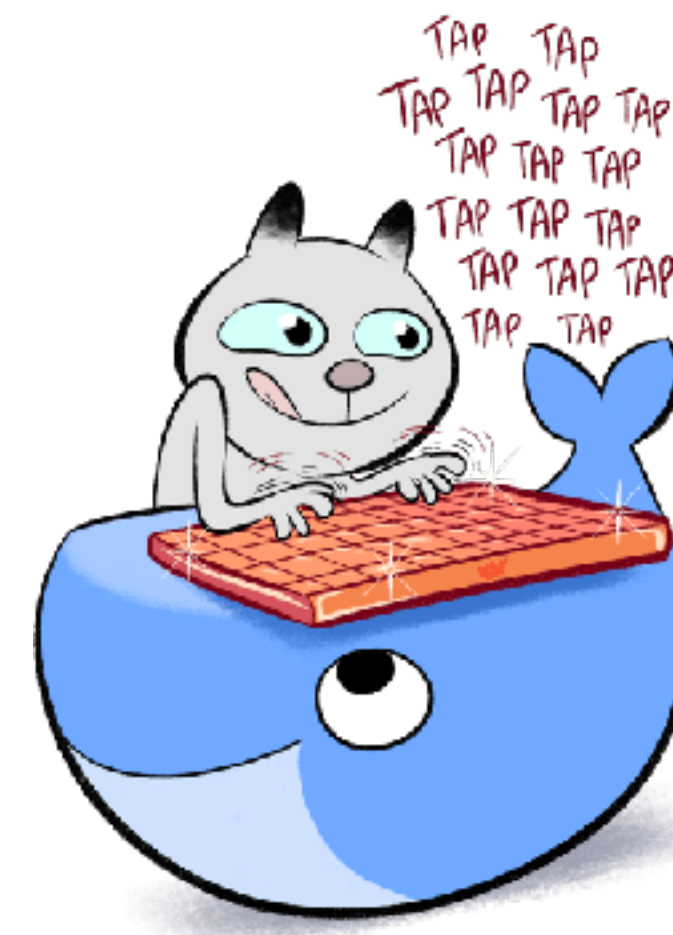
- Docker for Mac/Windows/Linux
- Native desktop or cloud provider experience
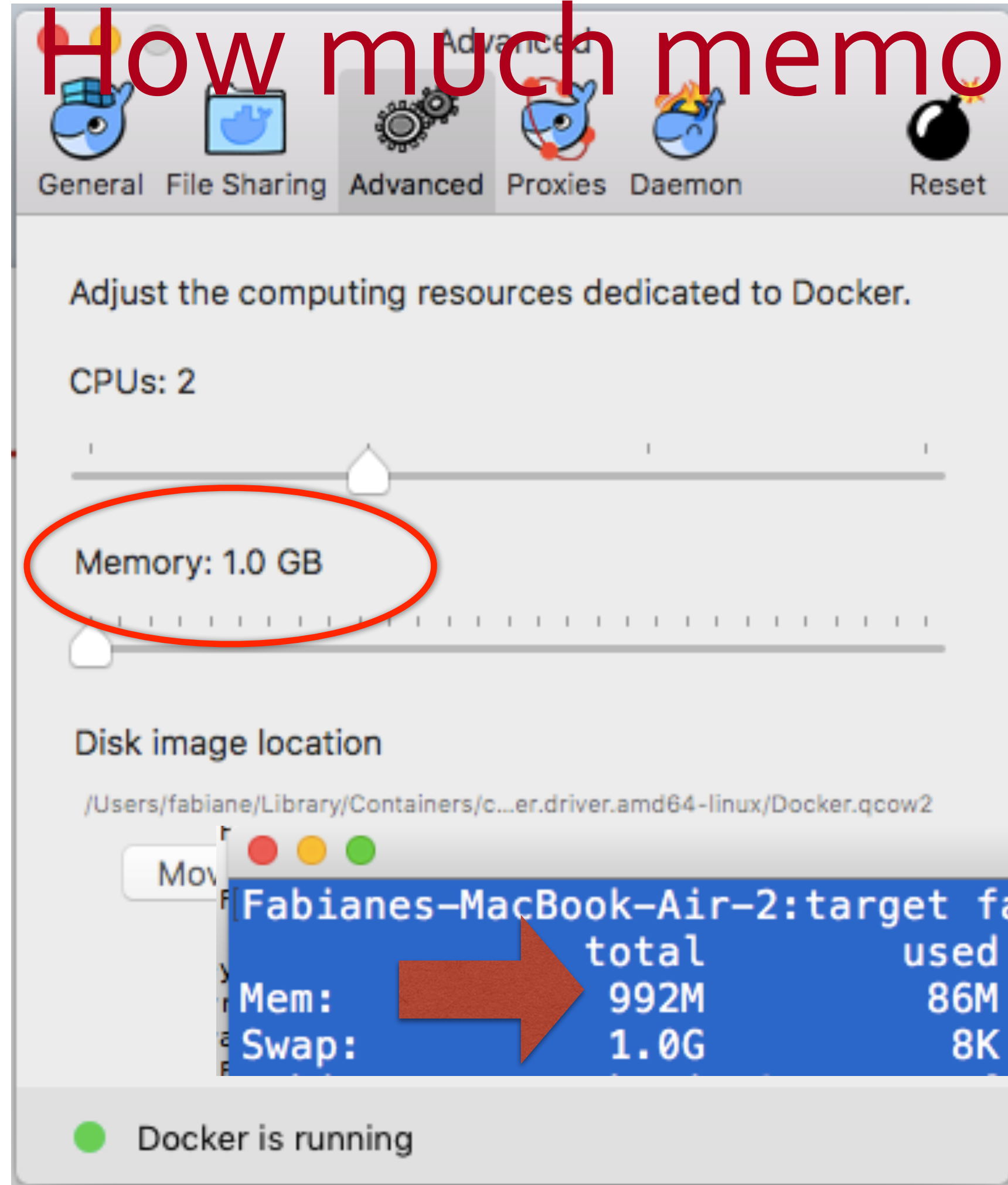- Monthly edge and quarterly stable release
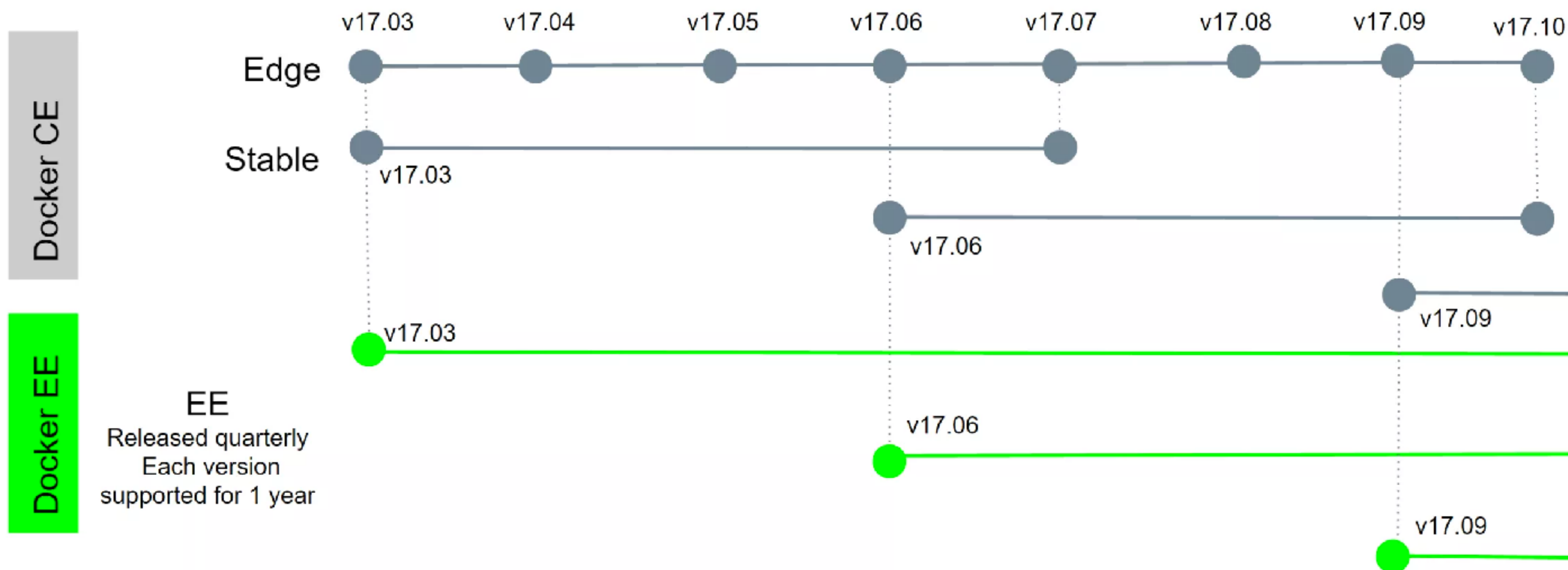
Docker for AWS/Azure

- Amazon Web Services
  - Amazon CloudFormation templates
  - Integrated with Autoscaling, ELB, and EBS
- Azure
  - Integrated with VM Scale Sets for autoscaling, Azure Load Balancer, Azure Storage
- Available in Docker CE and Docker EE

Memory Management

How much memory is available for containers?

How much memory is available for containers?

`$ docker run --memory=100M memory-sample`

docker memory switches instruct linux to kill the process if limits are exceeded

```
#2698842, Total: 253427712, Used: 148020032, Free: 105407680
#2727631, Total: 253427712, Used: 149402536, Free: 104025176
#2756421, Total: 253427712,                   102642688
#2785210, Total: 253427712,                   101260184
#2814000, Total: 253427712,                   99877696
#2842789, Total: 253427712,                   98495192
#2871579, Total: 253427712,                   97112704
#2900368, Total: 253427712,                   95730200
#2929158, Total: 253427712,                   94347712
#2957947, Total: 253427712,                   92965208
#2986737, Total: 253427712,                   91582720
#3015526, Total: 2534277   Used: 163227496, Free: 90200216
#3044316, Total: 2534277   Used: 164609984, Free: 88817728
#3073105, Total: 25342      Used: 165992488, Free: 87435224
#3101895, Total: 253427712, Used: 167374976, Free: 86052736
#3130684, Total: 253427712, Used: 168757480, Free: 84670232
Killed
```

—memory means RAM plus Swap

How much memory is available for containers?

# How much memory is available for containers?

- By default, container will use as much memory and swap

- Can be restricted using `-memory`, `-memory-reservation`, `-memory-swap`

- Today, JDK unaware of container's limited resources
  - For example, memory or CPU using `-cpus`, `-cpu-shares`

- JDK 9 has experimental support for cgroup memory limits
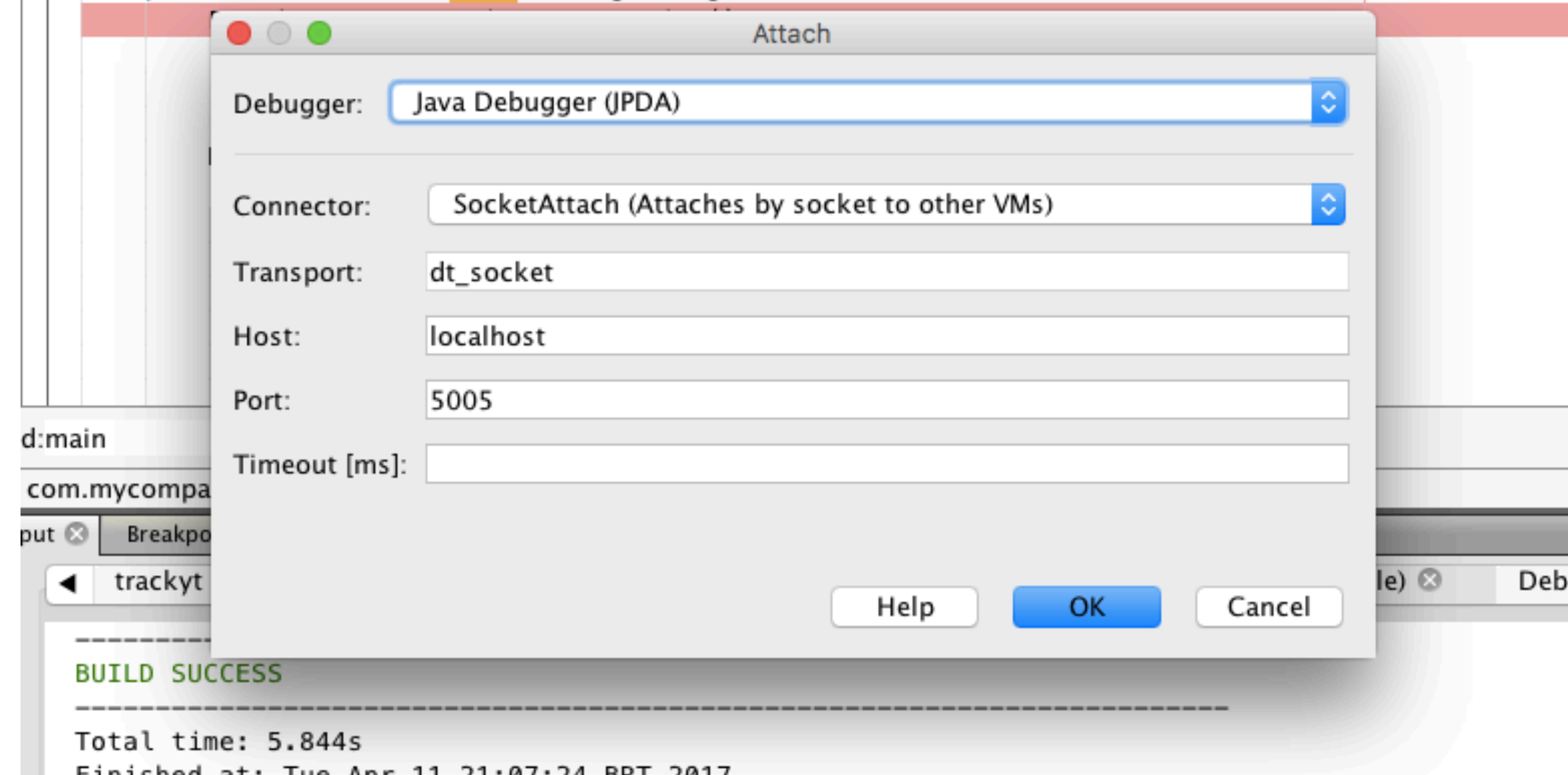
Debugging Java Applications

Running in debug mode

```
docker run -p5005:5005 \
   -e JAVA_OPTIONS='-Xdebug \
   -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=5005' \
   memory-sample
```

Attaching the IDE
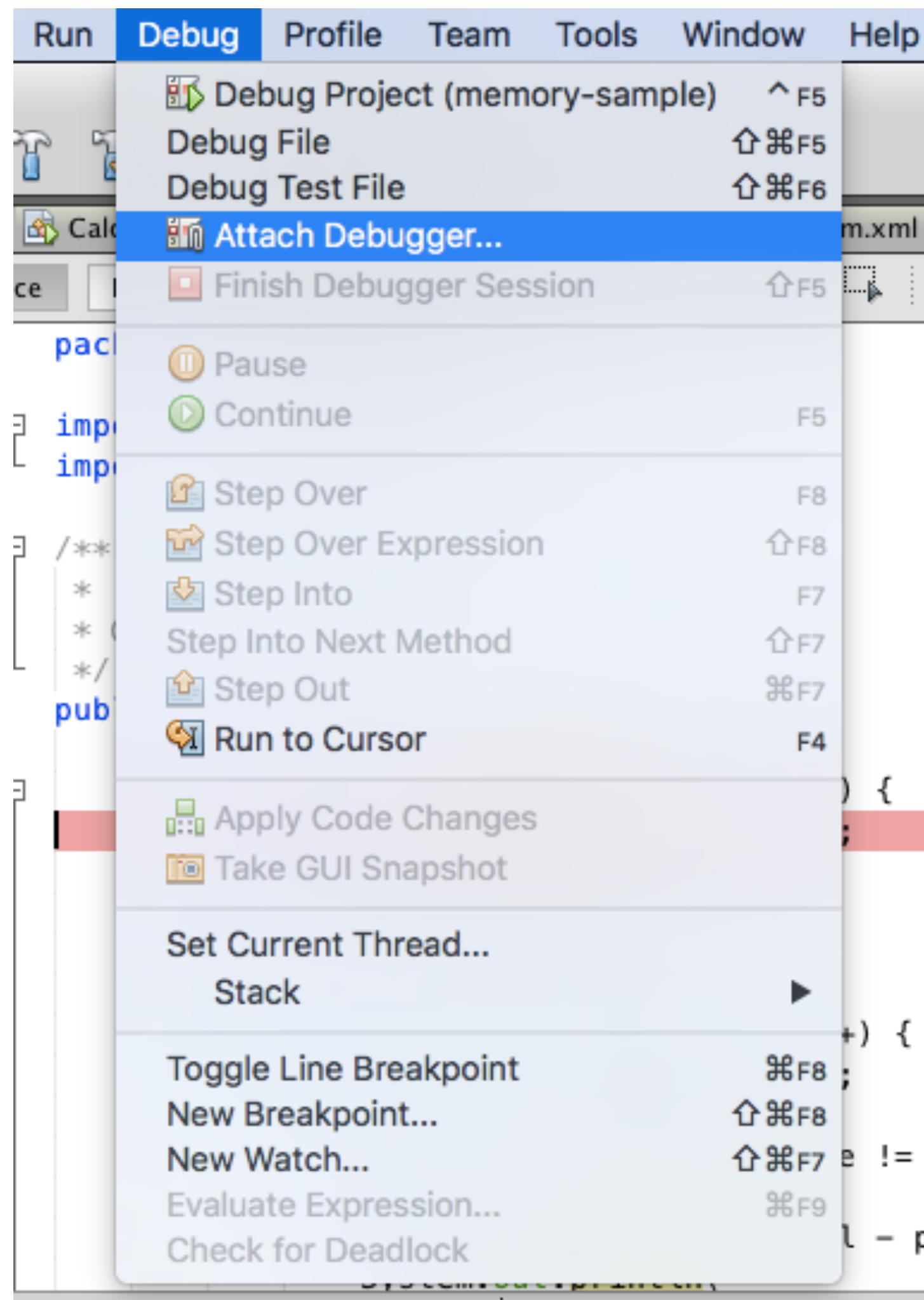
Monitor Java Applications

Monitoring Docker Containers

- **`docker stats`** command
  - LogEntries
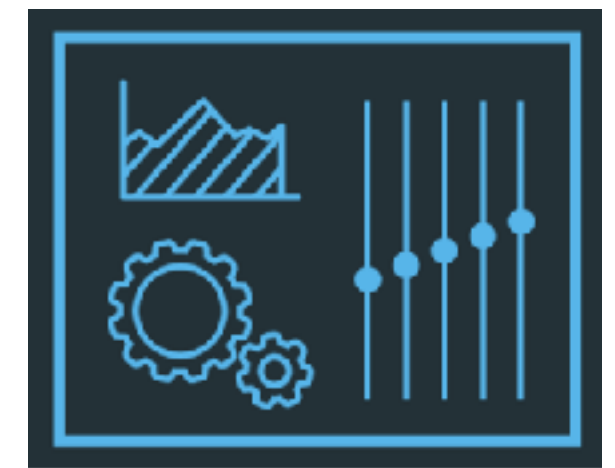- Service logs: `docker service logs <service>`
- Prometheus endpoint - New in 1.13
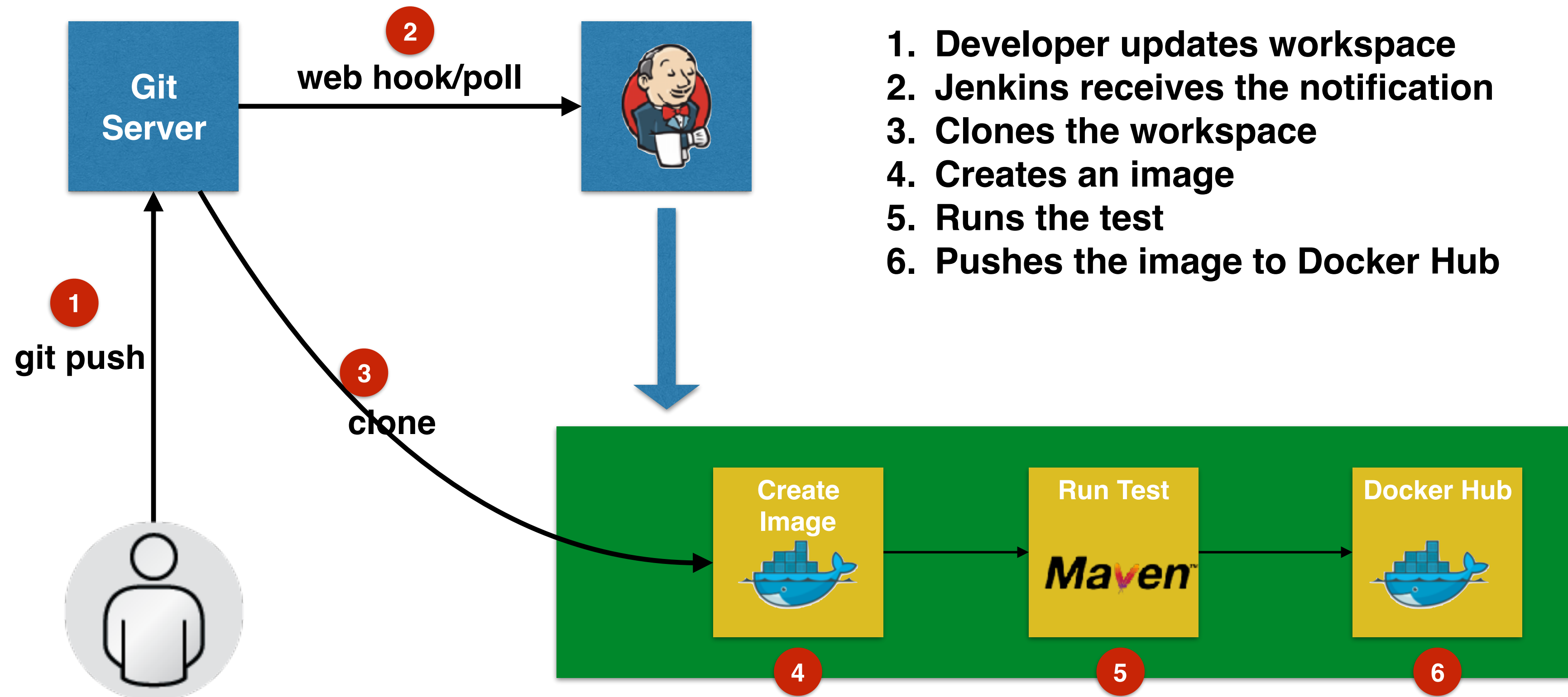- Docker Remote API: `/container/{container-name|cid}/stats`
- Docker Universal Control Plane
- cAdvisor
  - Prometheus
  - InfluxDB

CI/CD with Docker + Jenkins

1. Developer updates workspace
2. Jenkins receives the notification
3. Clones the workspace
4. Creates an image
5. Runs the test
6. Pushes the image to Docker Hub

http://blog.couchbase.com/2016/september/deployment-pipeline-docker-jenkins-java-couchbase

References

- Slides: github.com/docker/labs/tree/master/slides
- Workshop: github.com/docker/labs/tree/master/java
- Docs: docs.docker.com