Docker for Java Developers

Fabiane Nardon, @fabianenardon
Arun Gupta, @arungupta

Fabiane’s introduction

Docker Captain

Java Champion

JavaOne Rock Star (4 years)

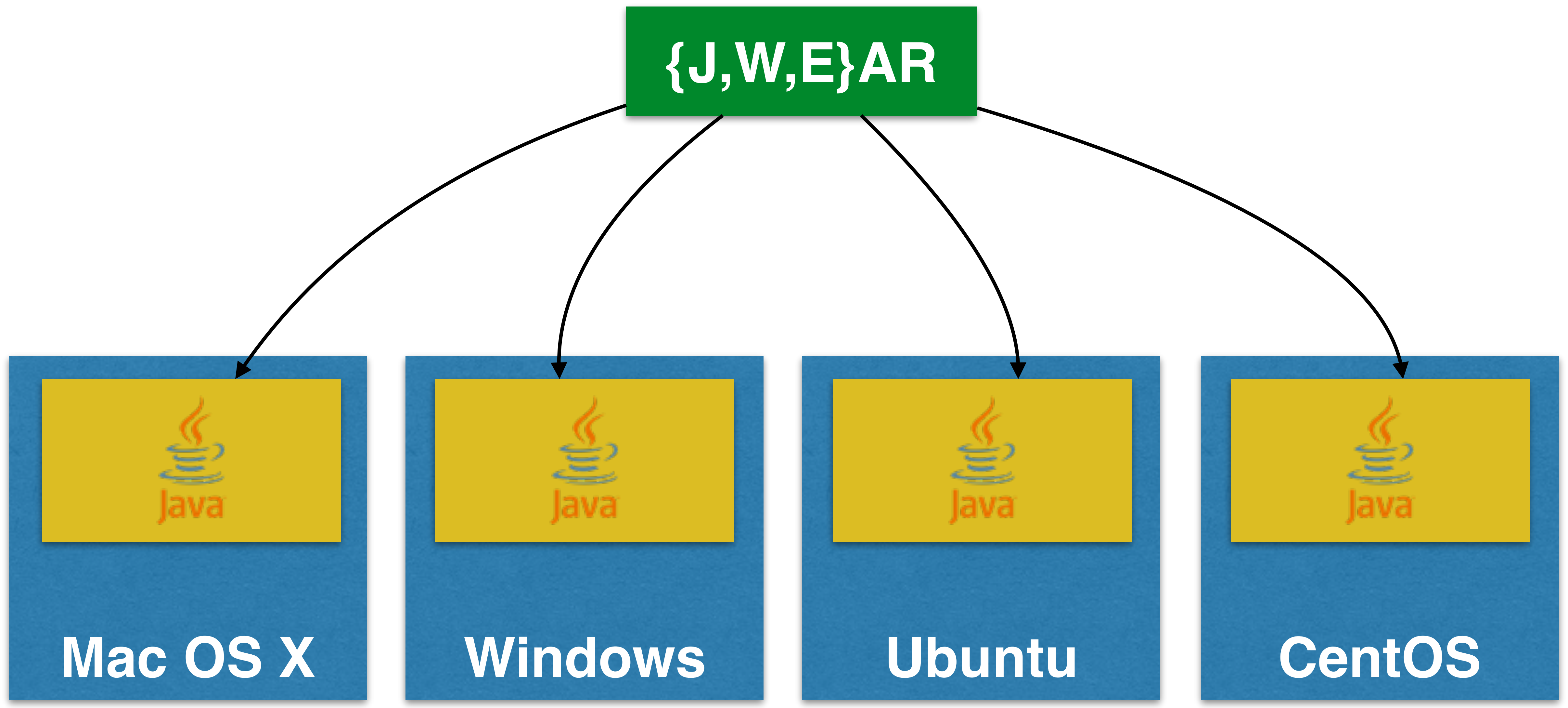
NetBeans Dream Team

Silicon Valley JUG Leader

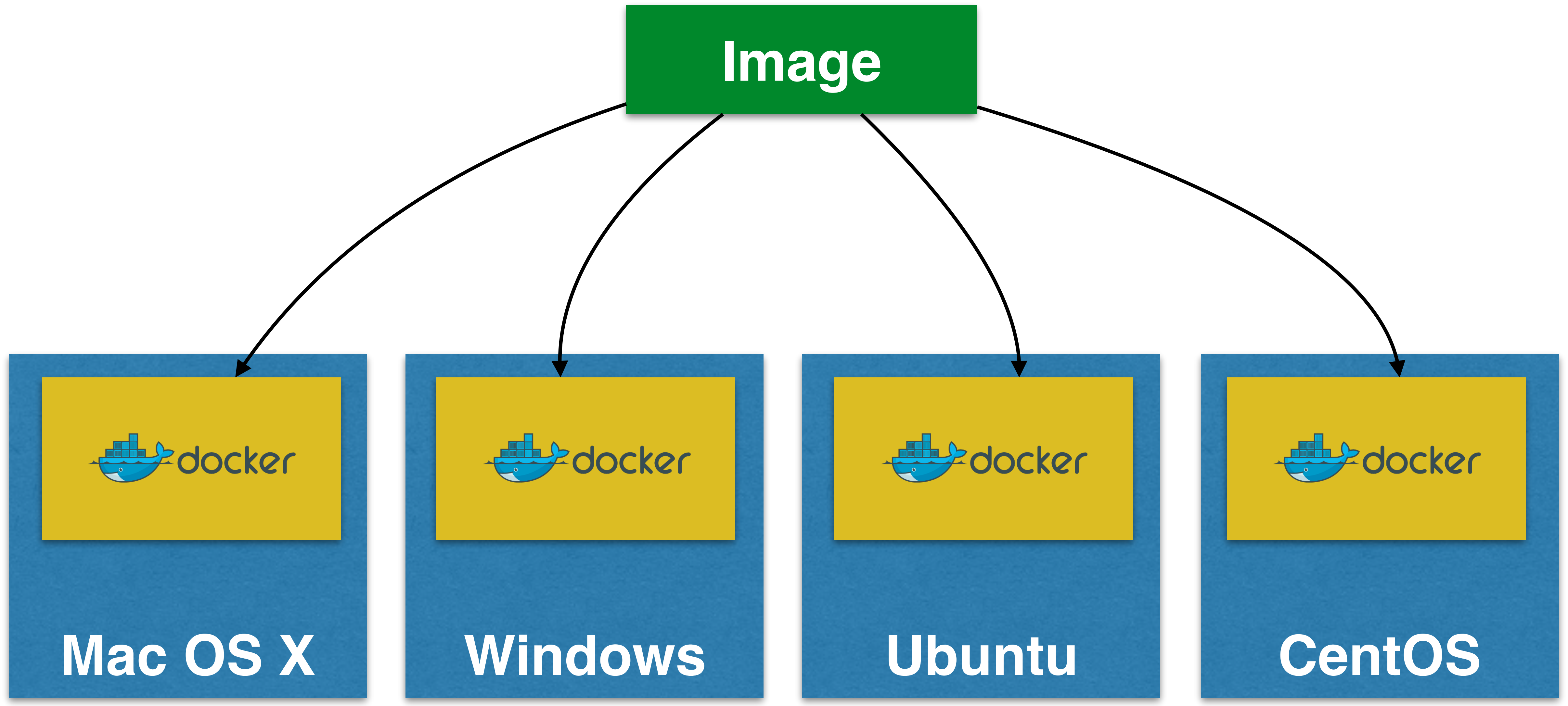
Author

Runner

Lifelong learner

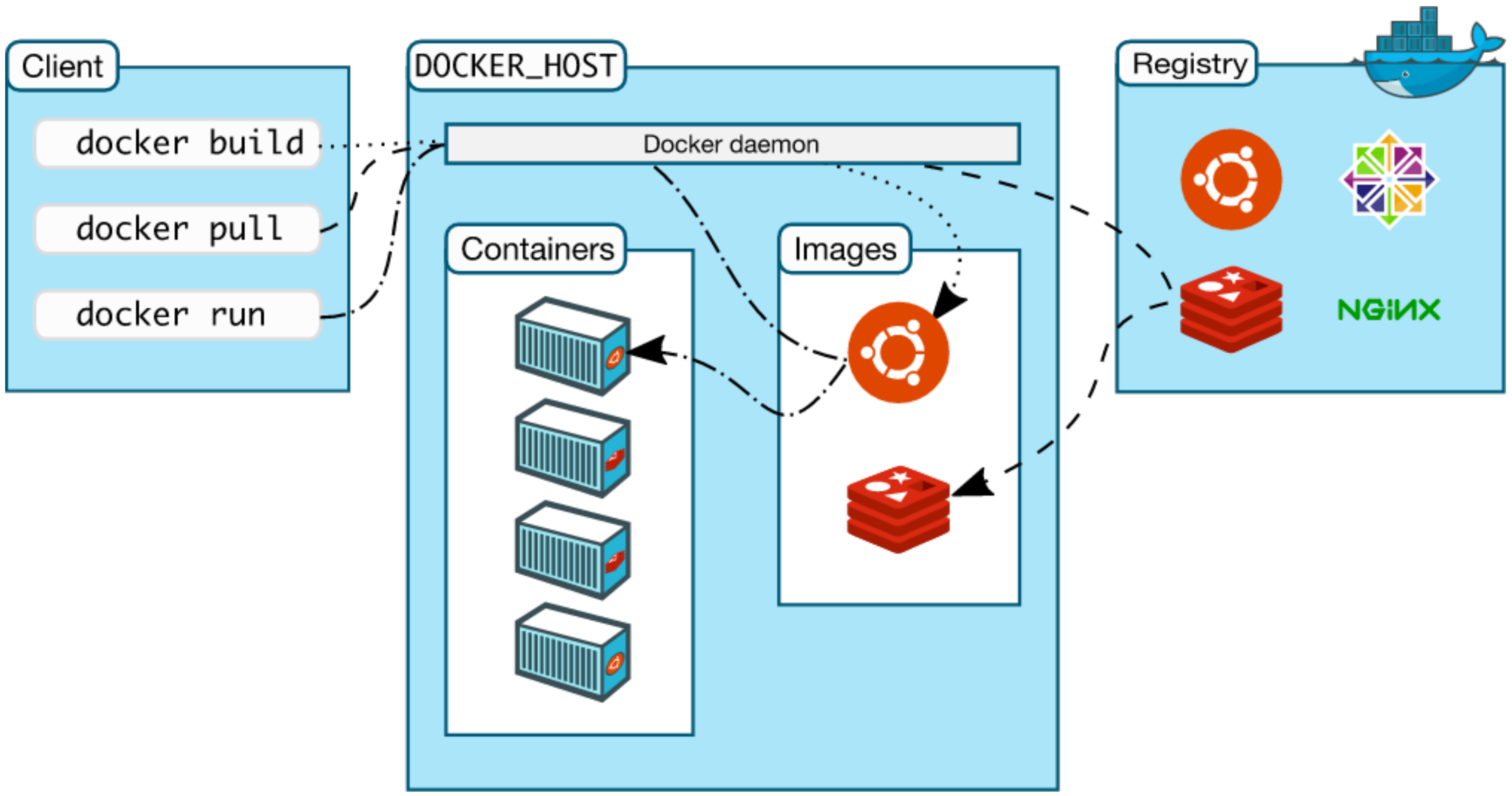


WORA = Write Once Run Anywhere



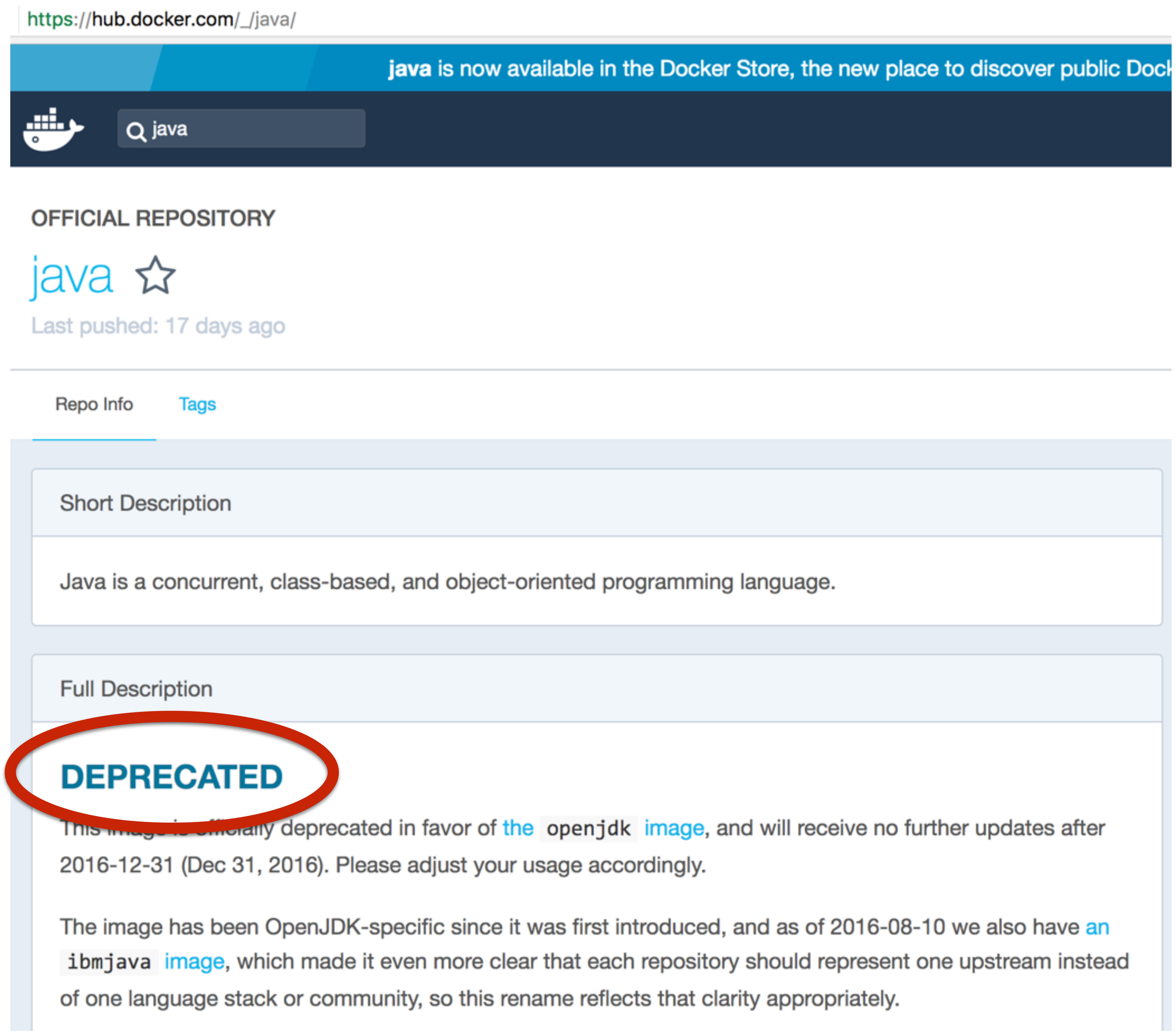
PODA = Package Once Deploy Anywhere

Docker Workflow



Java Base Image

Java base image #1



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

java is now available in the Docker Store, the new place to discover public Dock

java

OFFICIAL REPOSITORY

java

Last pushed: 17 days ago

Repo Info Tags

Short Description

Java is a concurrent, class-based, and object-oriented programming language.

Full Description

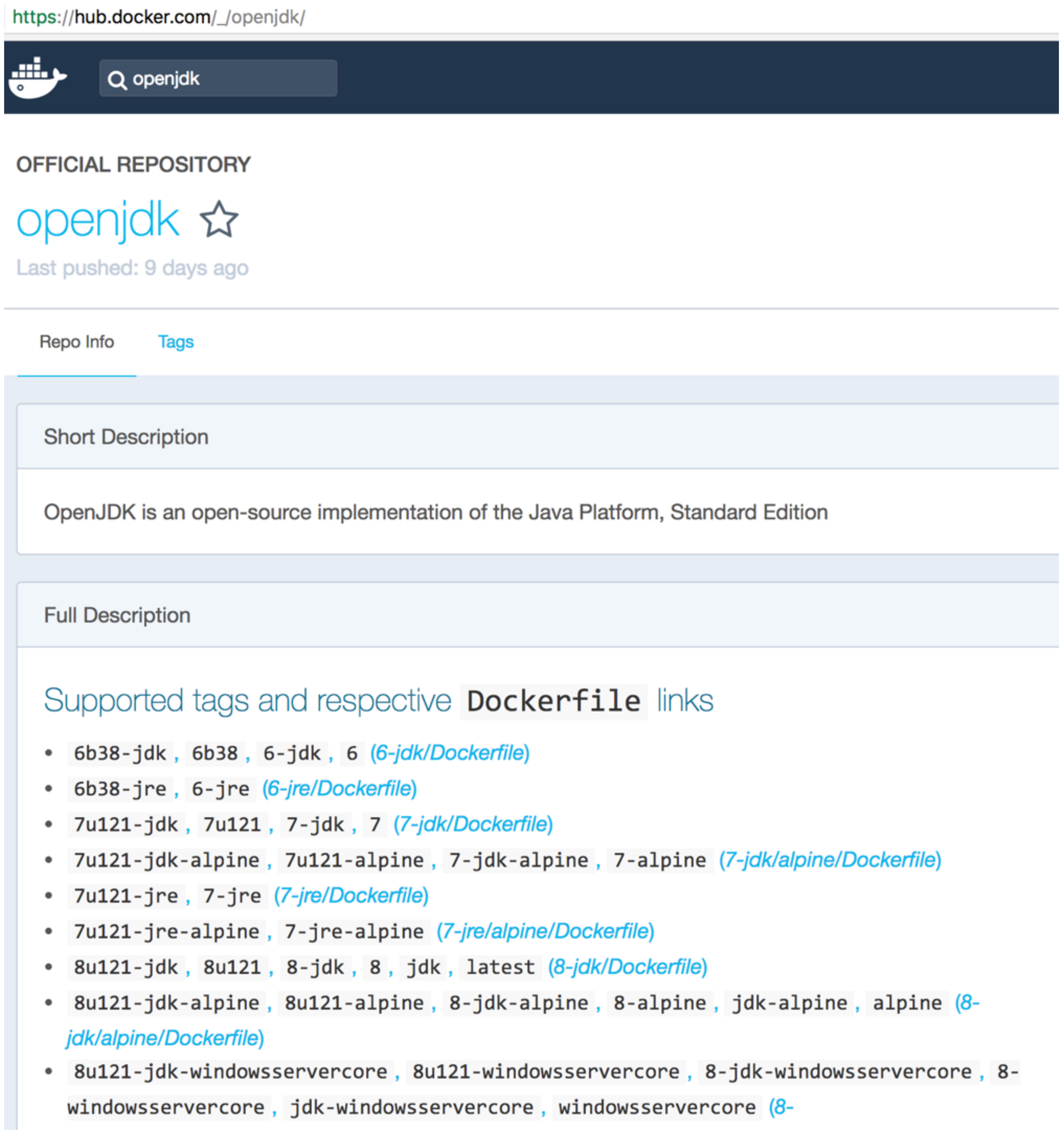
DEPRECATED

This image is officially deprecated in favor of the `openjdk` image, and will receive no further updates after 2016-12-31 (Dec 31, 2016). Please adjust your usage accordingly.

The image has been OpenJDK-specific since it was first introduced, and as of 2016-08-10 we also have an `ibmjava` image, which made it even more clear that each repository should represent one upstream instead of one language stack or community, so this rename reflects that clarity appropriately.

Java base image #2

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



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

openjdk

OFFICIAL REPOSITORY

openjdk

Last pushed: 9 days ago

Repo Info Tags

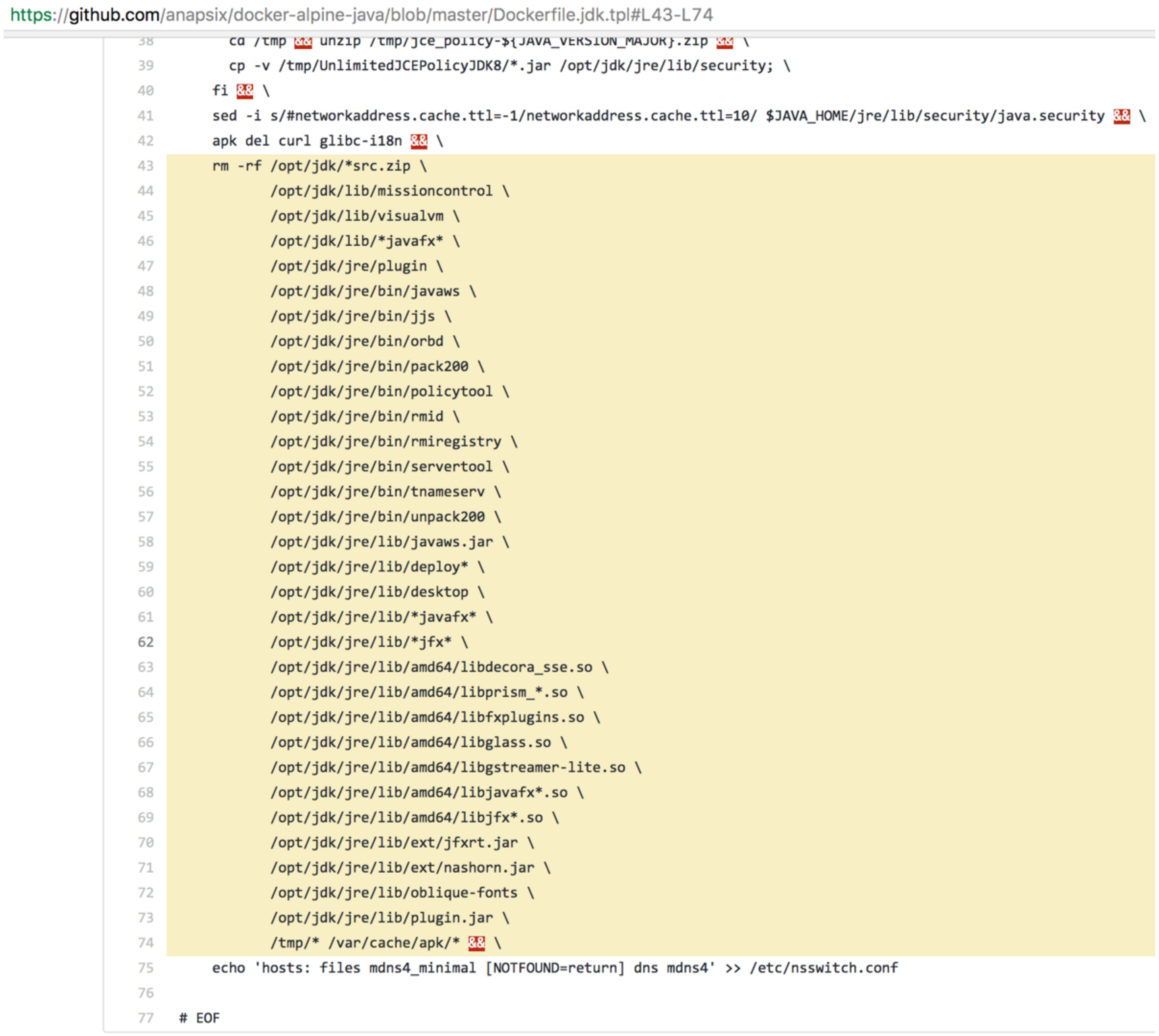
Short Description

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-

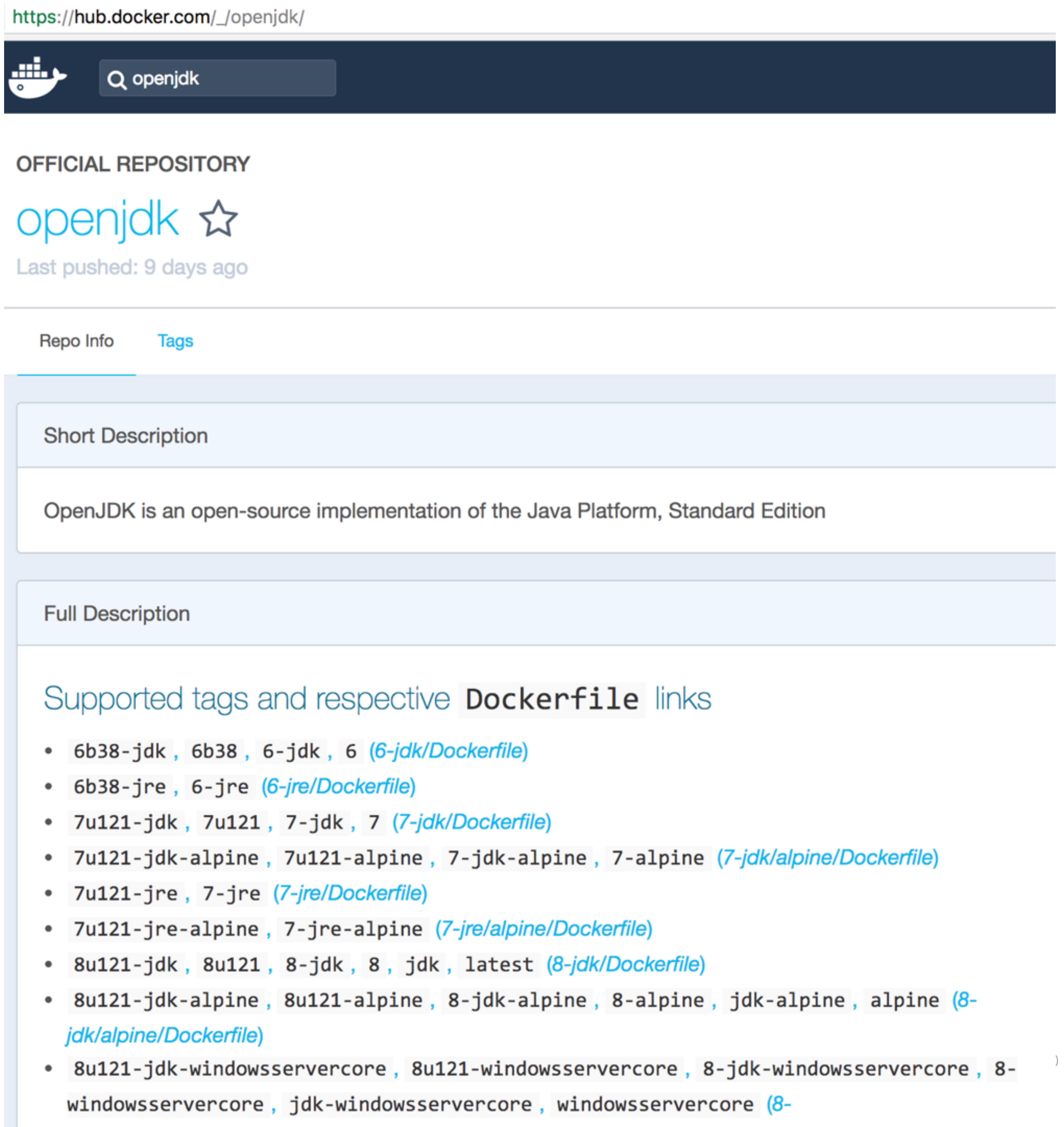


https://github.com/anapsix/docker-alpine-java/blob/master/Dockerfile.jdk.tpl#L43-L74

```
        cd /tmp && unzip /tmp/jce_policy-${JAVA_VERSION_MAJOR}.zip && \
        cp -v /tmp/UnlimitedJCEPolicyJDK8/*.jar /opt/jdk/jre/lib/security; \
      fi && \
      sed -i s/#networkaddress.cache.ttl=-1/networkaddress.cache.ttl=10/ $JAVA_HOME/jre/lib/security/java.security && \
      apk del curl glibc-i18n && \
      rm -rf /opt/jdk/*src.zip \
             /opt/jdk/lib/missioncontrol \
             /opt/jdk/lib/visualvm \
             /opt/jdk/lib/*javafx* \
             /opt/jdk/jre/plugin \
             /opt/jdk/jre/bin/javaws \
             /opt/jdk/jre/bin/jjs \
             /opt/jdk/jre/bin/orbd \
             /opt/jdk/jre/bin/pack200 \
             /opt/jdk/jre/bin/policytool \
             /opt/jdk/jre/bin/rmid \
             /opt/jdk/jre/bin/rmiregistry \
             /opt/jdk/jre/bin/servertool \
             /opt/jdk/jre/bin/tnameserv \
             /opt/jdk/jre/bin/unpack200 \
             /opt/jdk/jre/lib/javaws.jar \
             /opt/jdk/jre/lib/deploy* \
             /opt/jdk/jre/lib/desktop \
             /opt/jdk/jre/lib/*javafx* \
             /opt/jdk/jre/lib/*jfx* \
             /opt/jdk/jre/lib/amd64/libdecora_sse.so \
             /opt/jdk/jre/lib/amd64/libprism_*.so \
             /opt/jdk/jre/lib/amd64/libfxplugins.so \
             /opt/jdk/jre/lib/amd64/libglass.so \
             /opt/jdk/jre/lib/amd64/libgstreamer-lite.so \
             /opt/jdk/jre/lib/amd64/libjavafx*.so \
             /opt/jdk/jre/lib/amd64/libjfx*.so \
             /opt/jdk/jre/lib/ext/jfxrt.jar \
             /opt/jdk/jre/lib/ext/nashorn.jar \
             /opt/jdk/jre/lib/oblique-fonts \
             /opt/jdk/jre/lib/plugin.jar \
             /tmp/* /var/cache/apk/* && \
      echo 'hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4' >> /etc/nsswitch.conf

  # EOF
```

Java base image #3


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

OFFICIAL REPOSITORY

openjdk

Last pushed: 9 days ago

Short Description

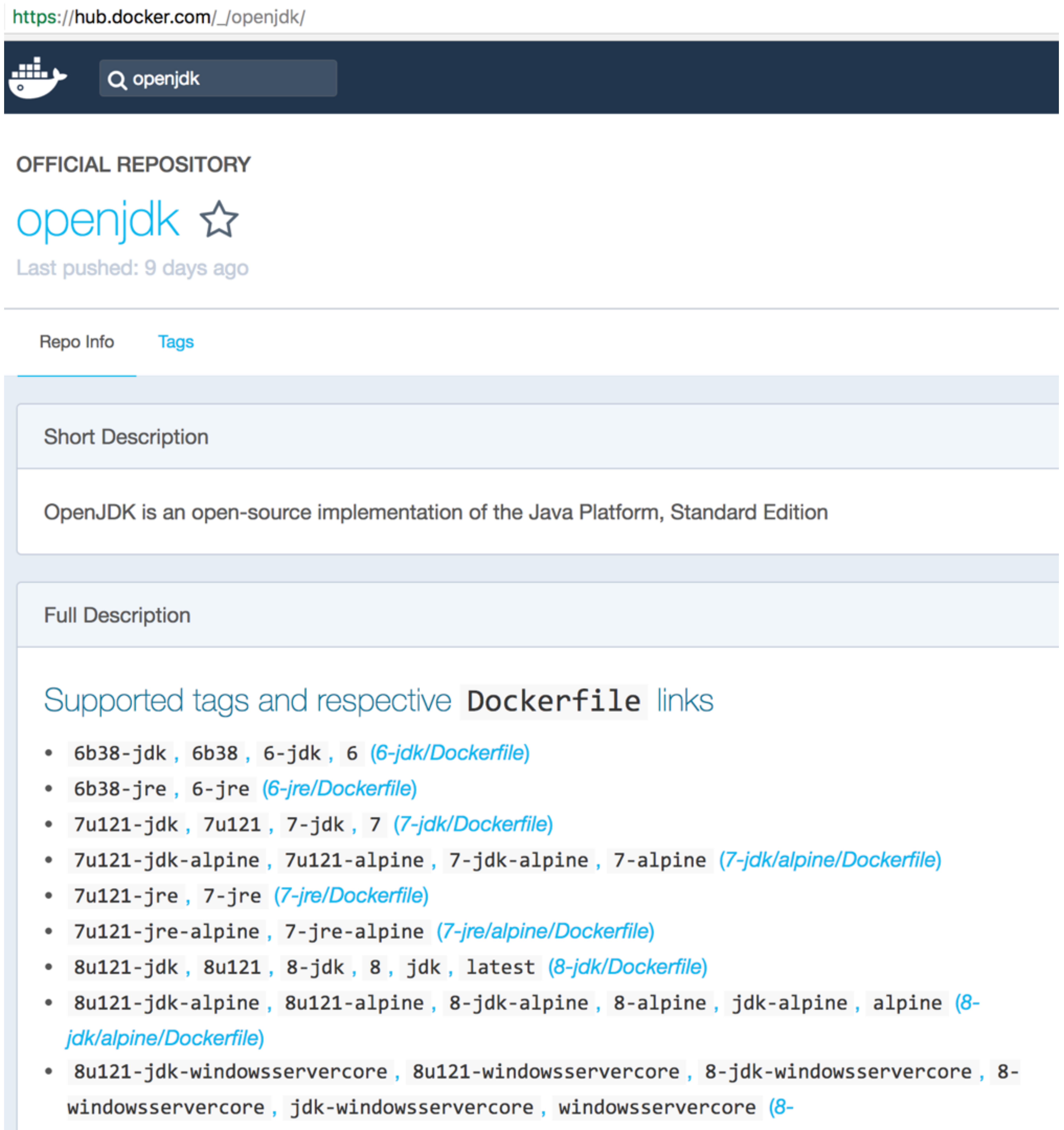
OpenJDK is an open-source implementation of the Java Platform, Standard Edition

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/

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

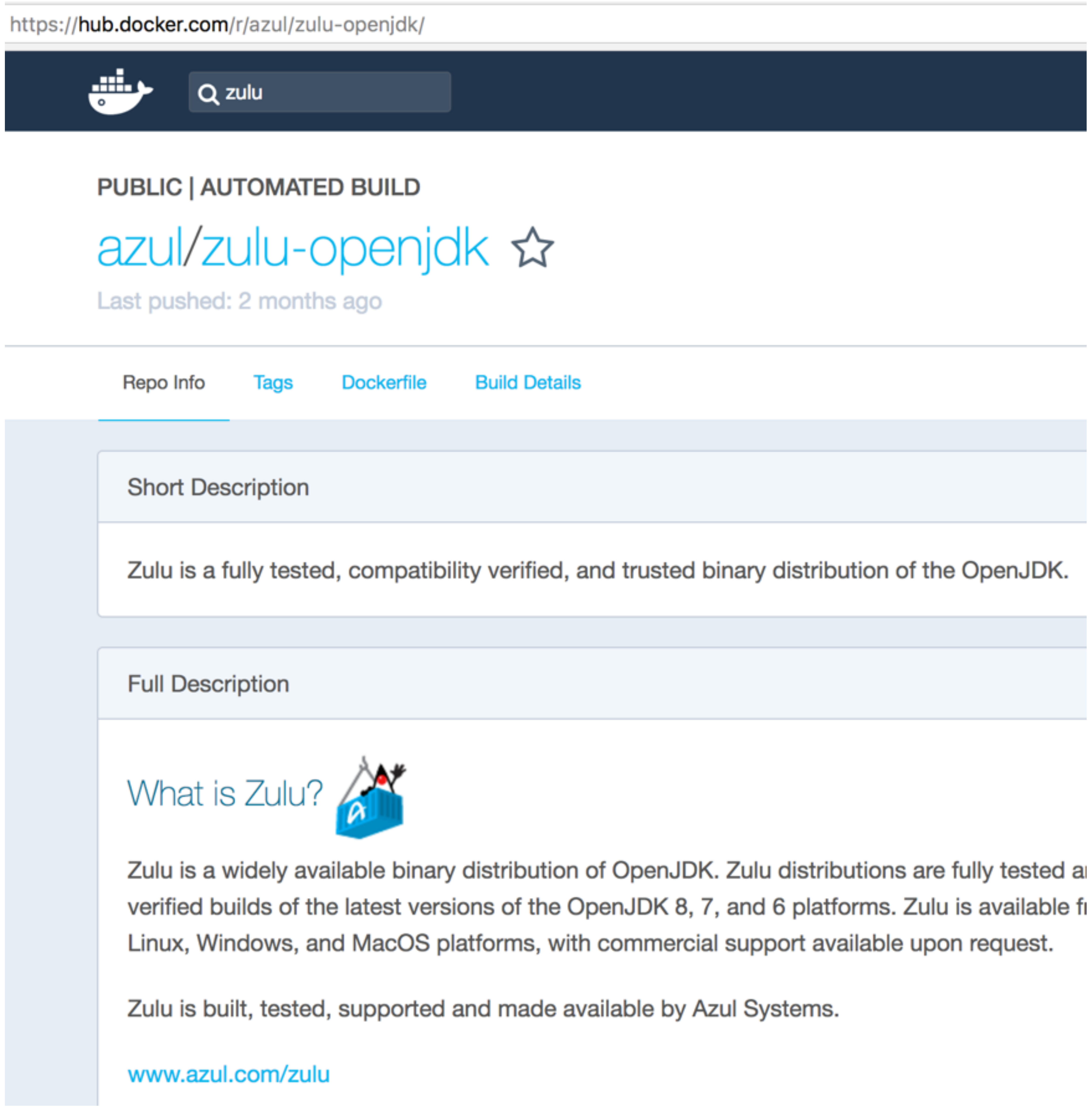
Full Description

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



PUBLIC | AUTOMATED BUILD

azul/zulu-openjdk

Last pushed: 2 months ago

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

First Java Docker Image

```
FROM openjdk:jdk-alpine

CMD java -version
```

First Java Web App Docker Image

```
FROM jboss/wildfly:10.1.0.Final

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

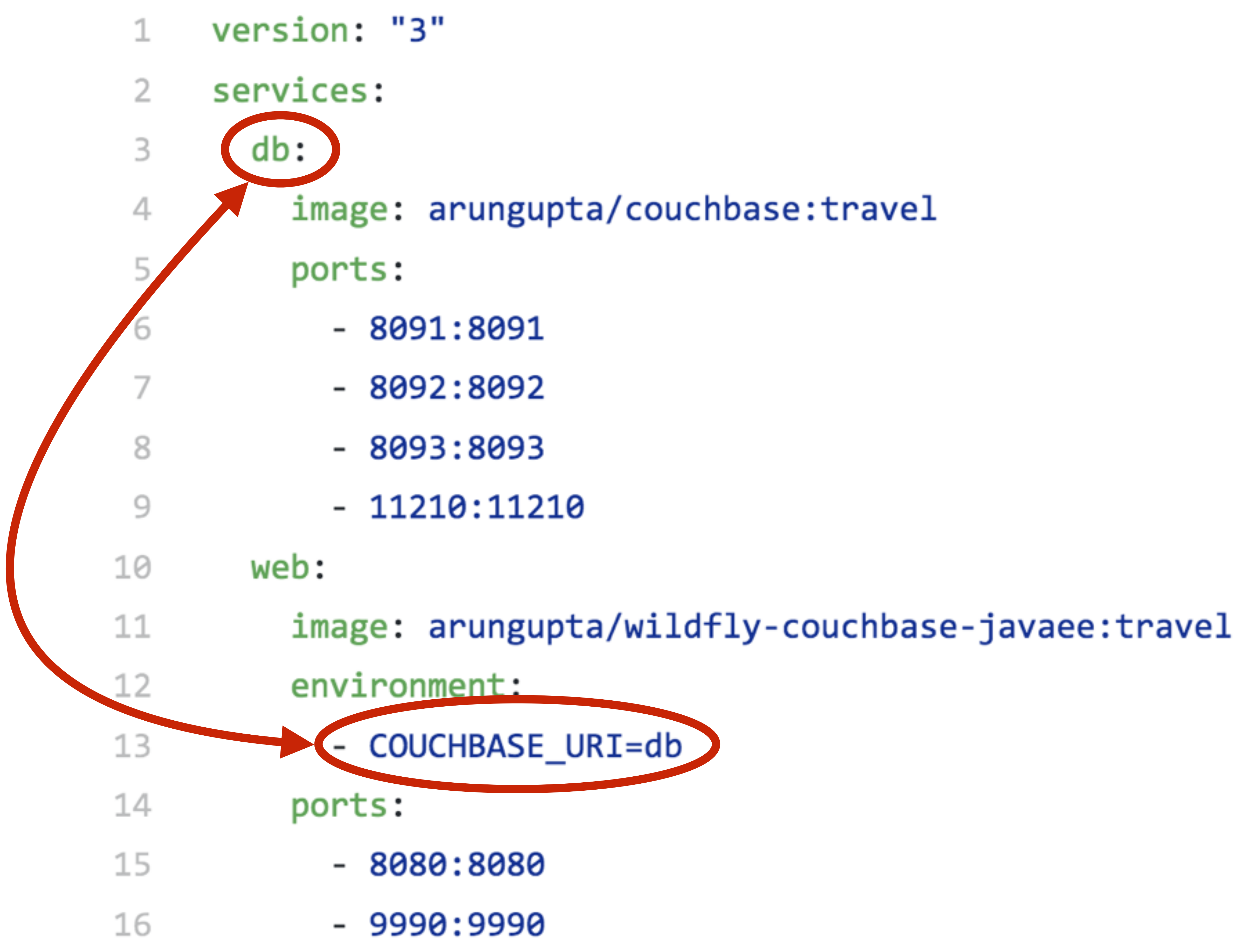
Package Docker + Java Application using Maven or Gradle

Multi-Container Application using Docker Compose

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`
- Deployed as Docker Stack
- Great for dev, staging, and CI

Service Discovery with Docker



```
version: "3"
services:
  db:
    image: arungupta/couchbase:travel
    ports:
      - 8091:8091
      - 8092:8092
      - 8093:8093
      - 11210:11210
  web:
    image: arungupta/wildfly-couchbase-javaee:travel
    environment:
      - COUCHBASE_URI=db
    ports:
      - 8080:8080
      - 9990:9990
```

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

Docker 1.13 - Compose v3

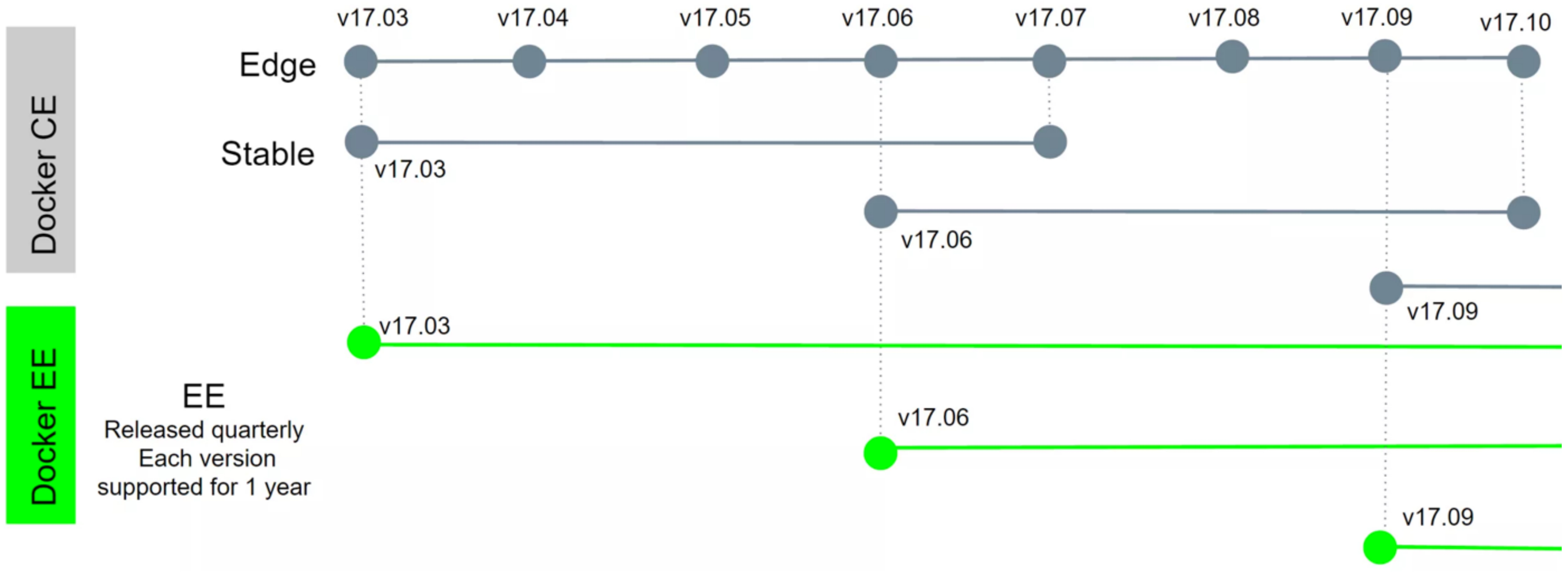
- `docker stack deploy` now supports Compose file
  - Number of desired instances of each service
  - Rolling update
  - Server constraints

Multi-Container Application on Multi-Host using Docker for AWS

Development: Docker

TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP TAP

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



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
- docker.com/getdocker

Docker for Mac/Windows

- Native application and UI
- Auto update capability
- No additional software required, e.g. VirtualBox
  - OSX: xhyve VM using `Hypervisor.framework`
  - Windows: Hyper-V VM
- Download: docker.com/getdocker
- Requires Yosemite 10.10+ or Windows 10 64-bit

Monitor Docker + Java Applications

Docker Compose Common Use Cases

Use Case	Command
Dev Setup	`docker-compose up`
Local/remote host	`DOCKER_HOST, DOCKER_TLS_VERIFY, DOCKER_CERT_PATH`
Single/multiple hosts	Integrated with Swarm
Multiple isolated environments	`docker-compose up -p <project>`

Automated test setup	`docker-compose up`
	`mvn test`
	`docker-compose down`

Dev/Prod Impedance mismatch	`docker-compose up -f docker-compose.yml -f production.yml`

Docker 1.13

- Deploy Compose services to Swarm
- CLI restructured
- Clean-up commands
- Monitoring commands
- Build improvements
- Improved CLI backwards compatibility
- Docker for AWS/Azure for Production

Docker 1.13 - CLI Restructured

```
Management Commands:
  checkpoint  Manage checkpoints
  container   Manage containers
  image       Manage images
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
  swarm       Manage Swarm
  system      Manage Docker
  volume      Manage volumes
```

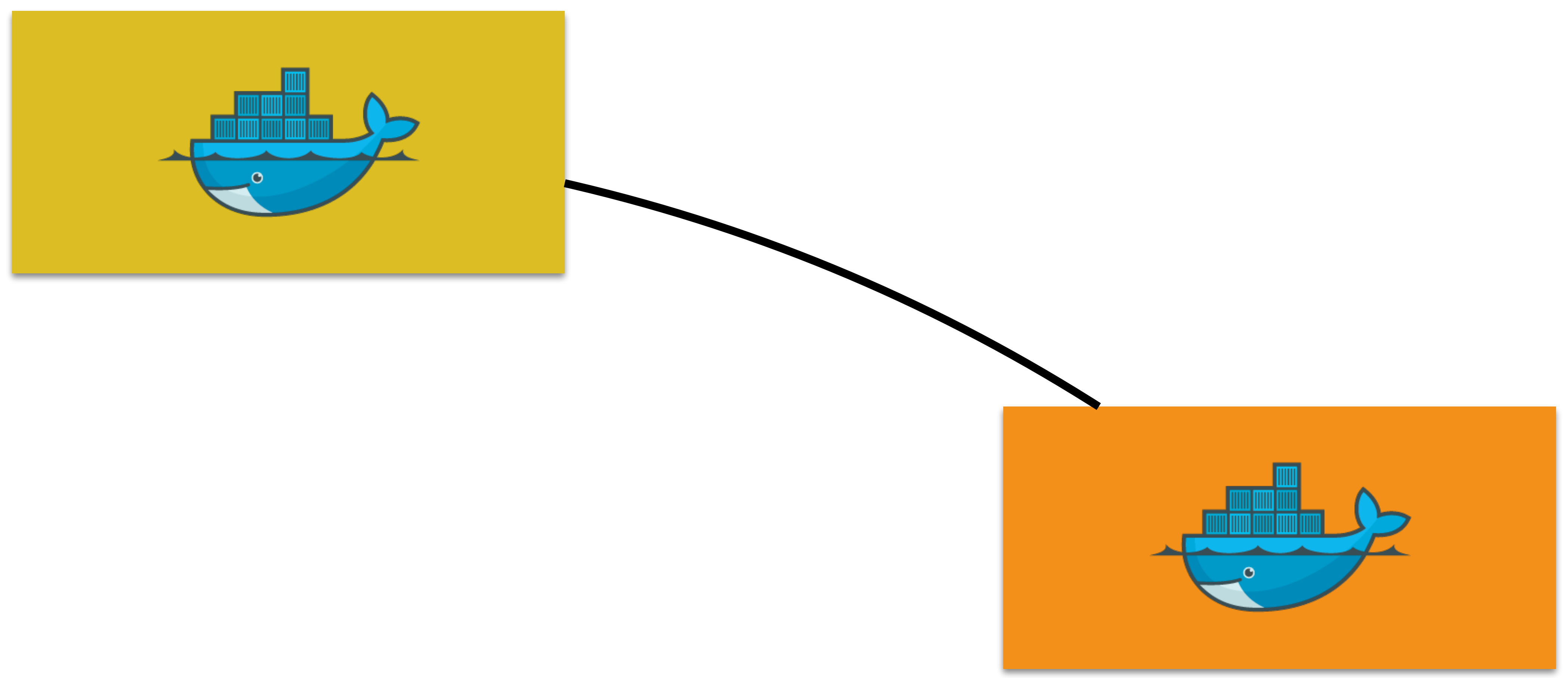
Swarm Mode

- New in 1.12
- 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: Initialize

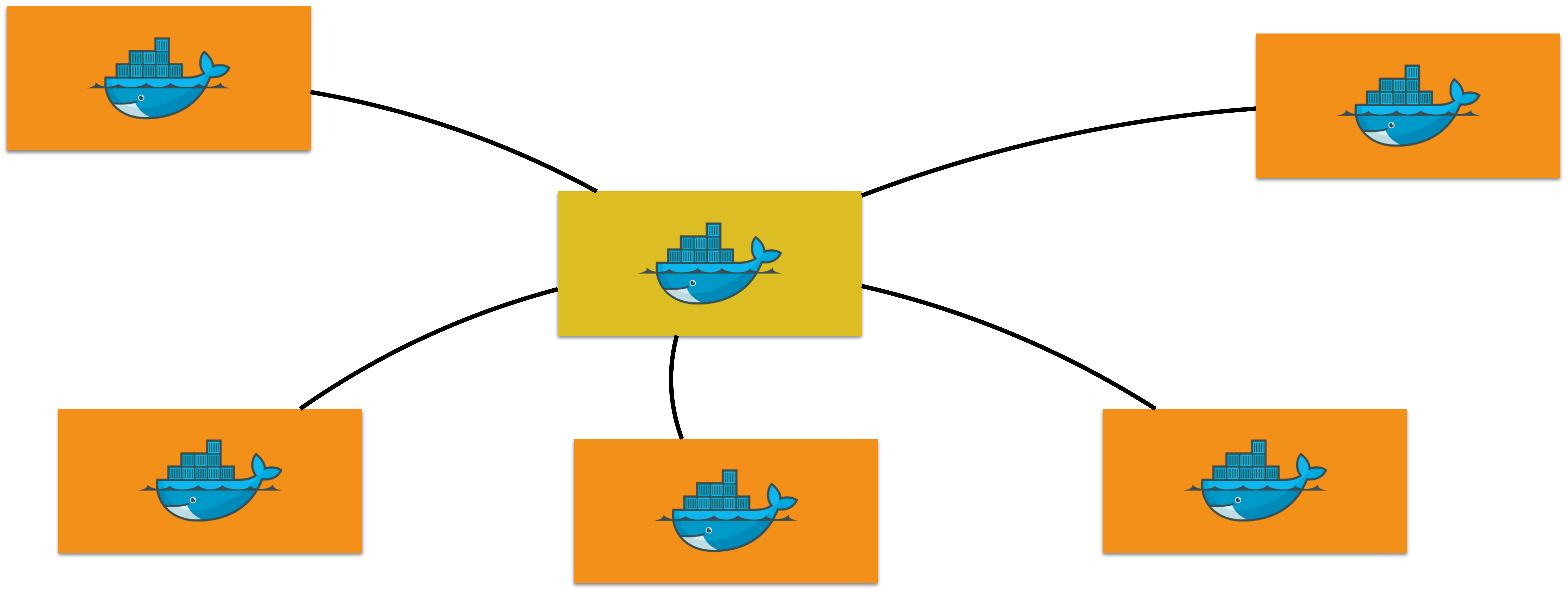
```
docker swarm init --listen-addr <ip>:2377
```

Swarm Mode: Add Worker



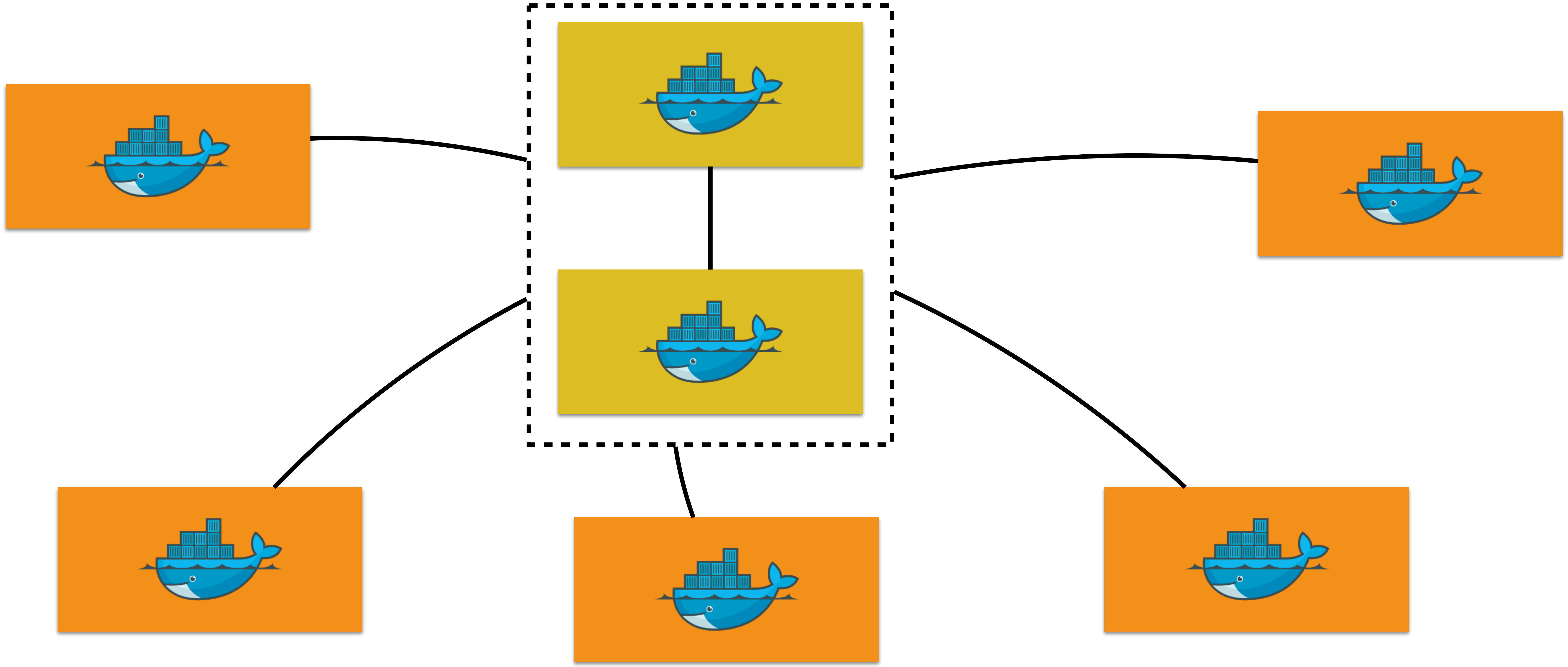
```
docker swarm join --token <worker_token> <manager>:2377
```

Swarm Mode: Add More Workers



```
docker swarm join --token <worker_token> <manager>:2377
```

Swarm Mode: Primary/Secondary Master



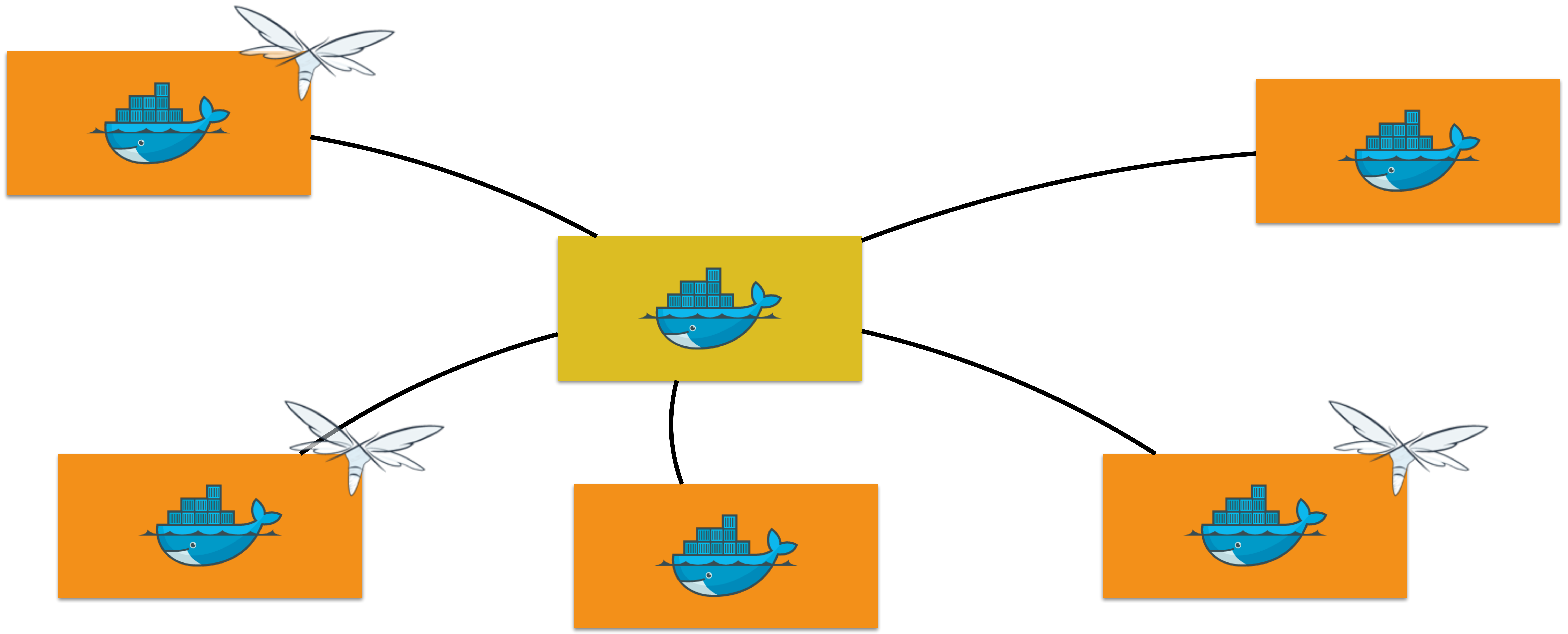
```
docker swarm join --manager --token <manager_token> --listen-
addr <master2>:2377 <master1>:2377
```

Swarm Mode using Docker Machine

Task	Command
Create manger	`docker-machine create -d virtualbox managerX`
Create worker	`docker-machine create -d virtualbox workerX`
Initialize Swarm mode	`docker swarm init --listen-addr <ip1> --advertise-addr <ip1>`
Manager token	`docker swarm join-token manager -q`
Worker token	`docker swarm join-token worker -q`
Manager X join	`docker swarm join --token manager_token --listen-addr <ipX> --advertise-addr <ipX> <ip1>`
Worker X join	`docker swarm join --token worker_token --listen-addr <ipX> --advertise-add <ipX> <ip1>`

https://github.com/docker/labs/blob/master/swarm-mode/quickstart/buildswarm-node-vbox-setup.sh

Swarm Mode: Replicated Service

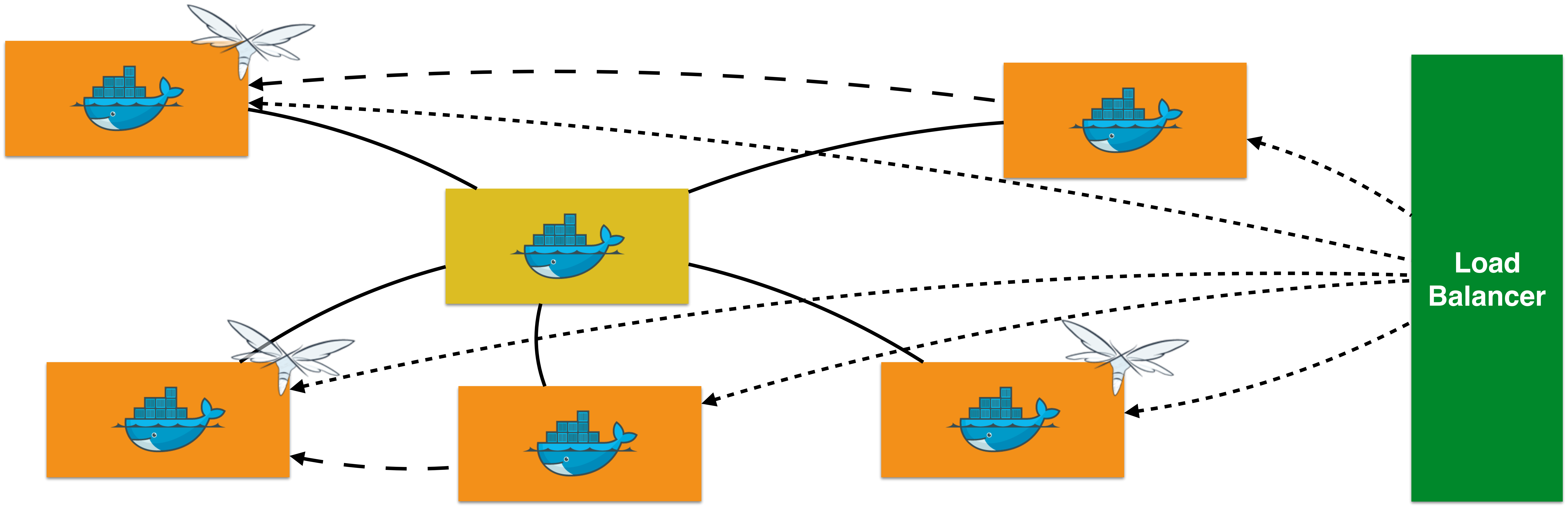


```
docker service create --replicas 3 --name web jboss/wildfly
```

Swarm Mode - Routing Mesh

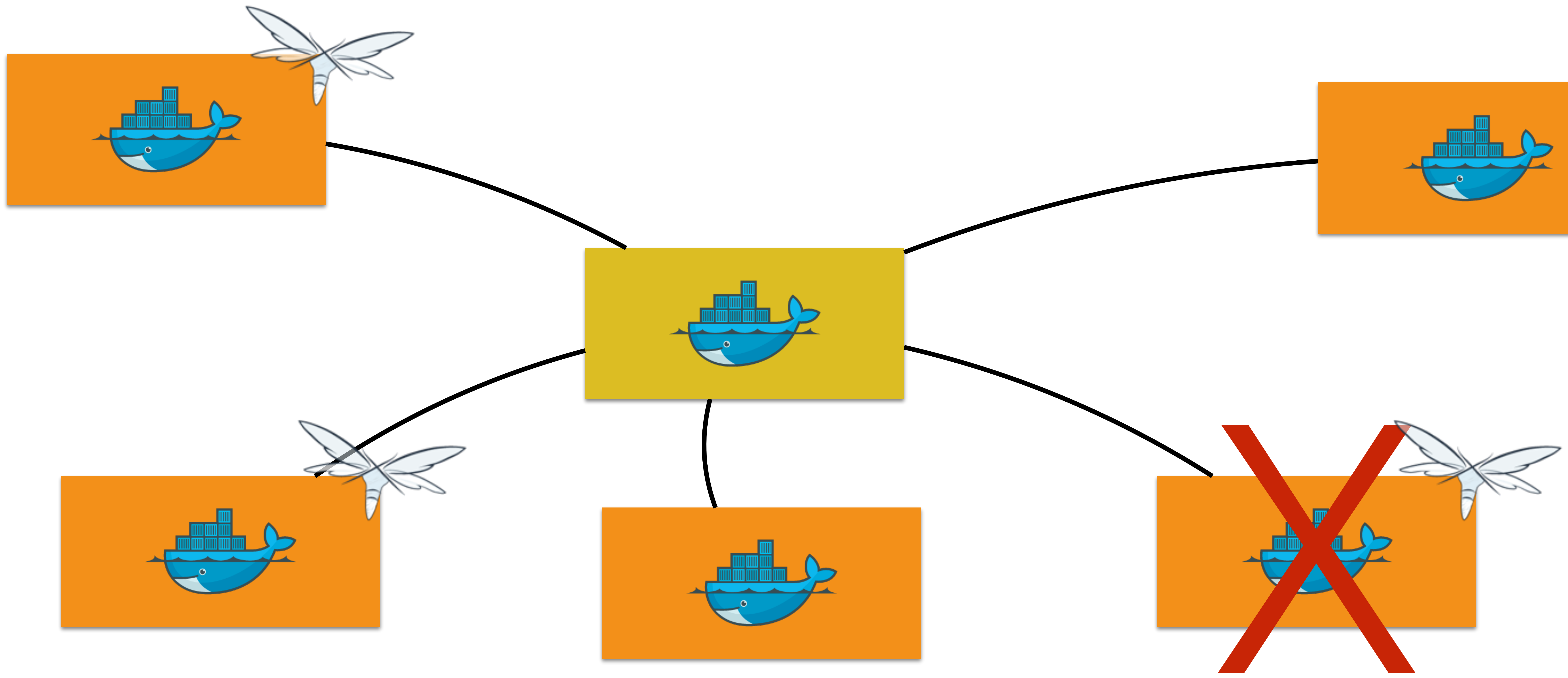
- Load balancers are host-aware, not container-aware
- Swarm mode introduces container-aware routing mesh
- Reroutes traffic from any host to a container
  - Reserves a Swarm-wide ingress port
  - Uses DNS-based service discovery

Swarm Mode: Routing Mesh

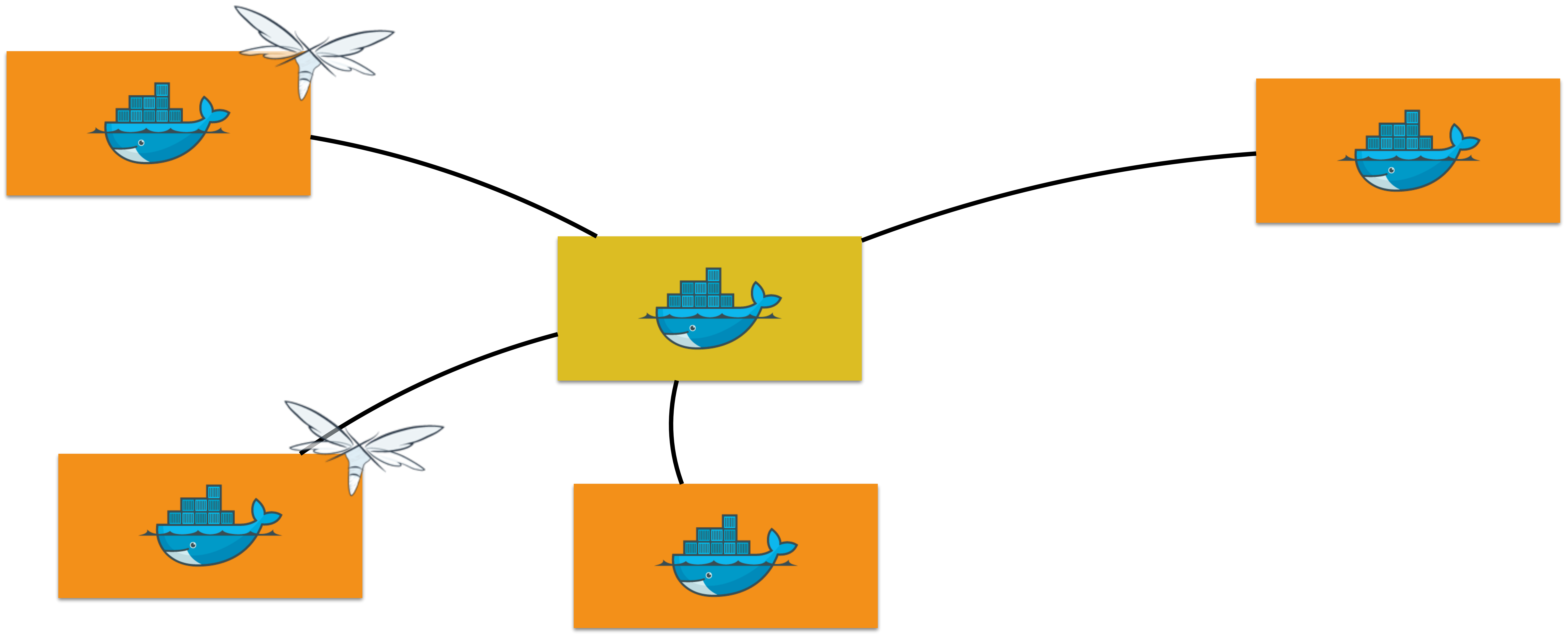


```
docker service create --replicas 3 --name web -p 8080:8080 jboss/
wildfly
```

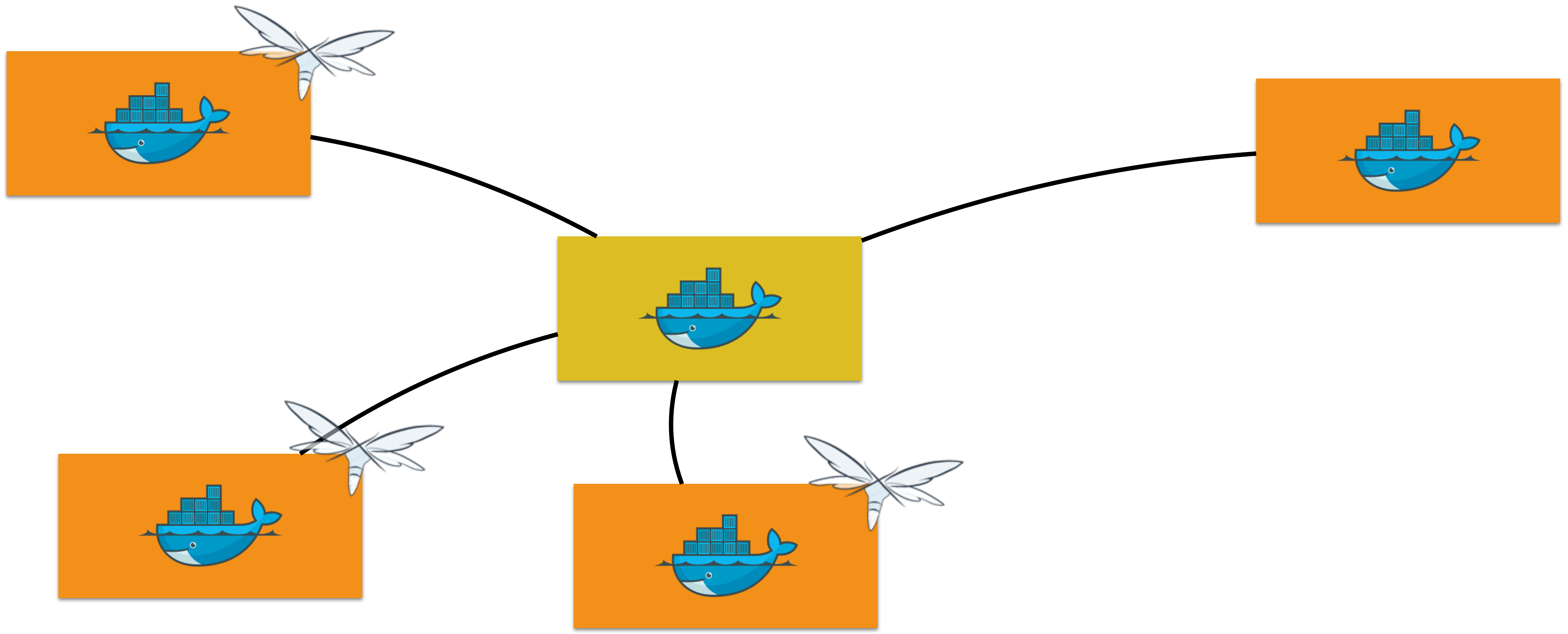
Swarm Mode: Node Failure



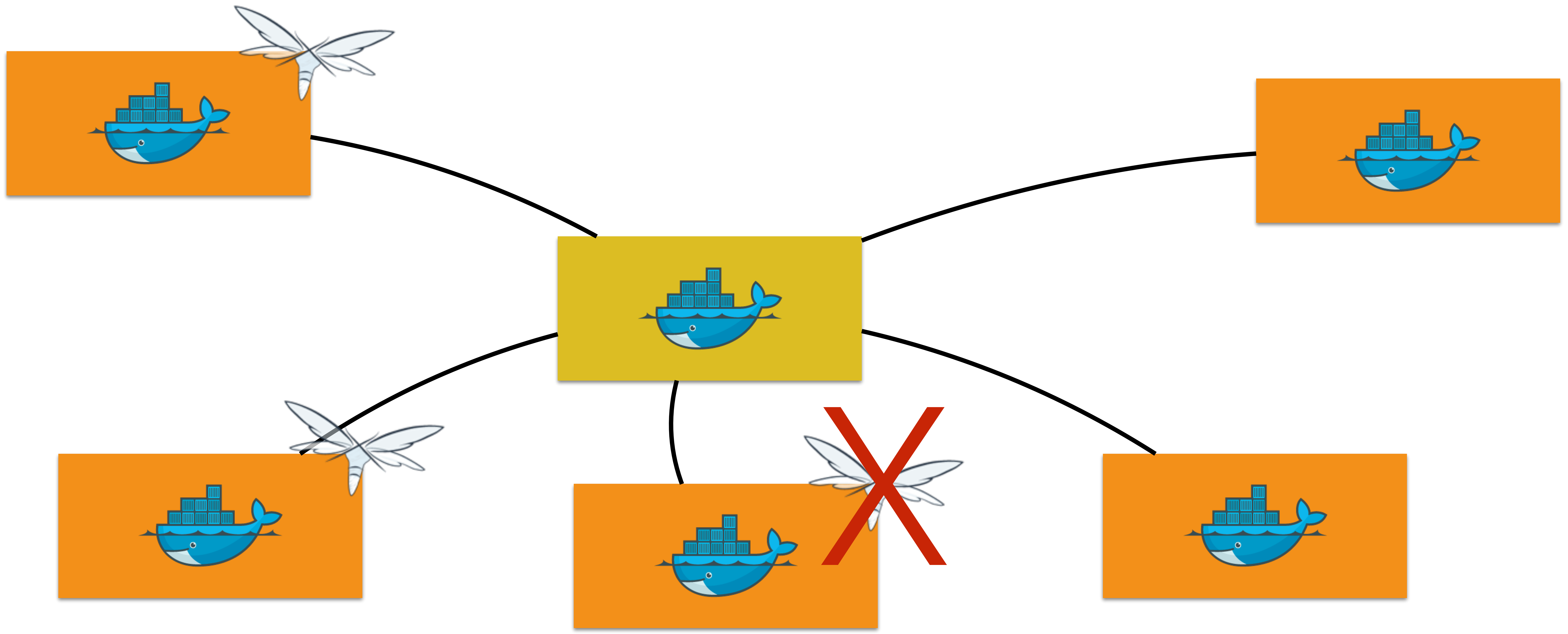
Swarm Mode: Desired != Actual



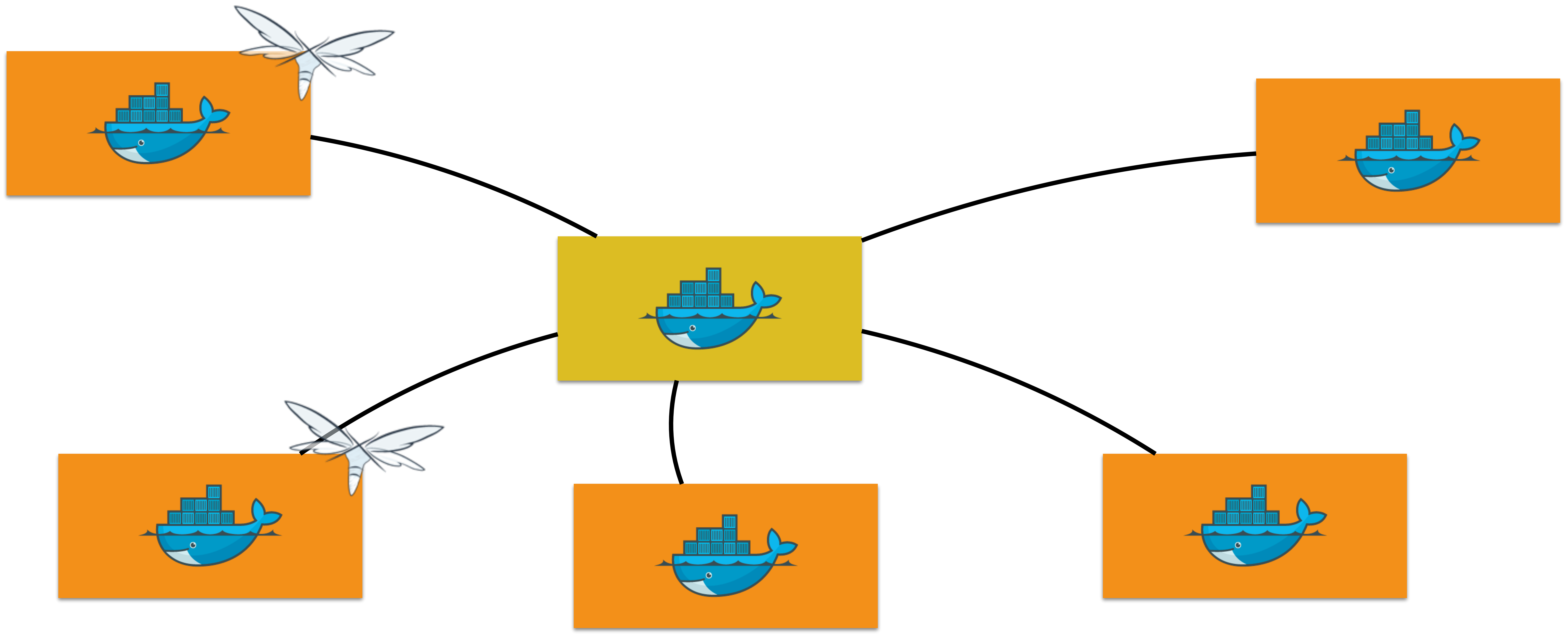
Swarm Mode: Reconcile



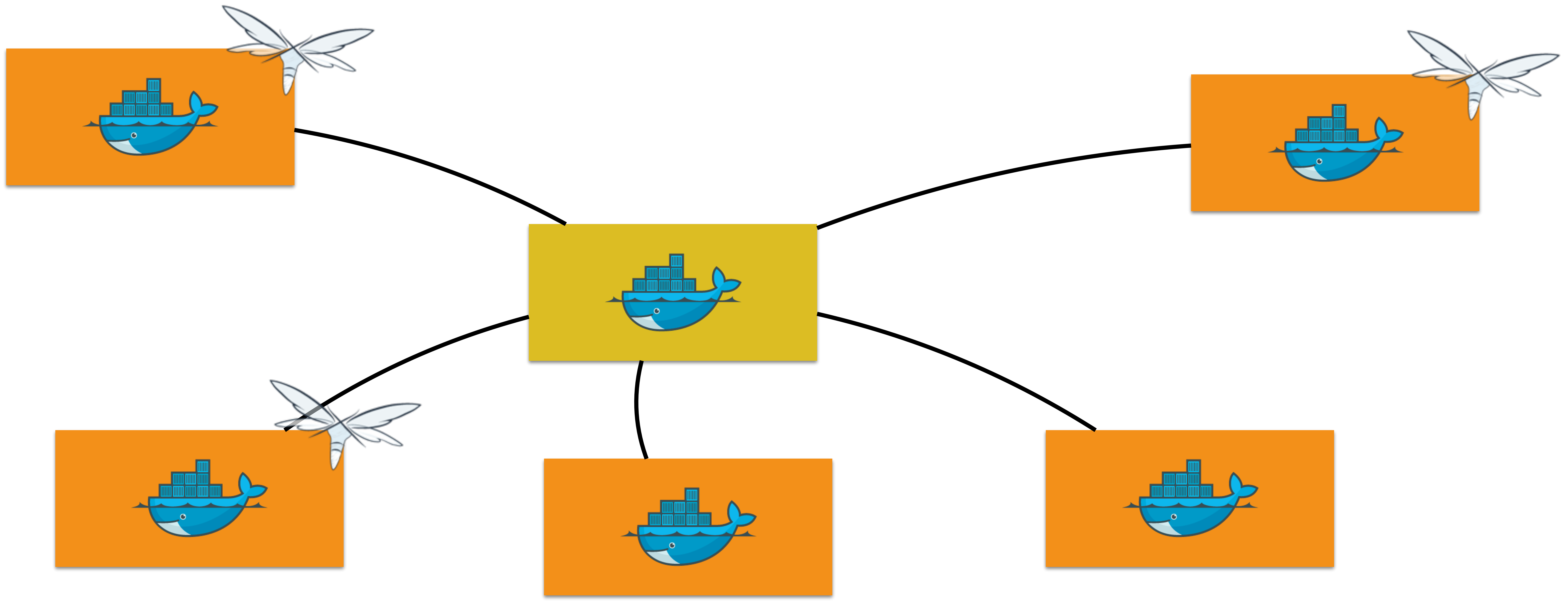
Swarm Mode: Container Failure



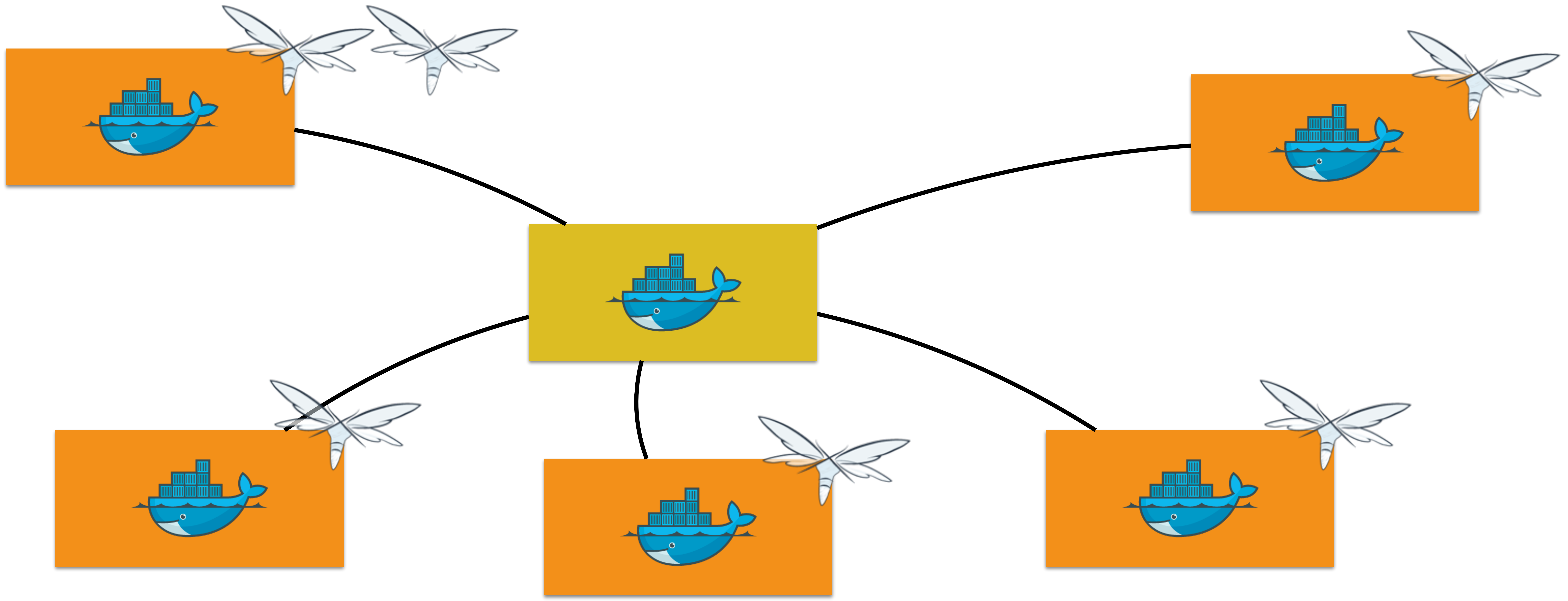
Swarm Mode: Desired != Actual



Swarm Mode: Reconcile

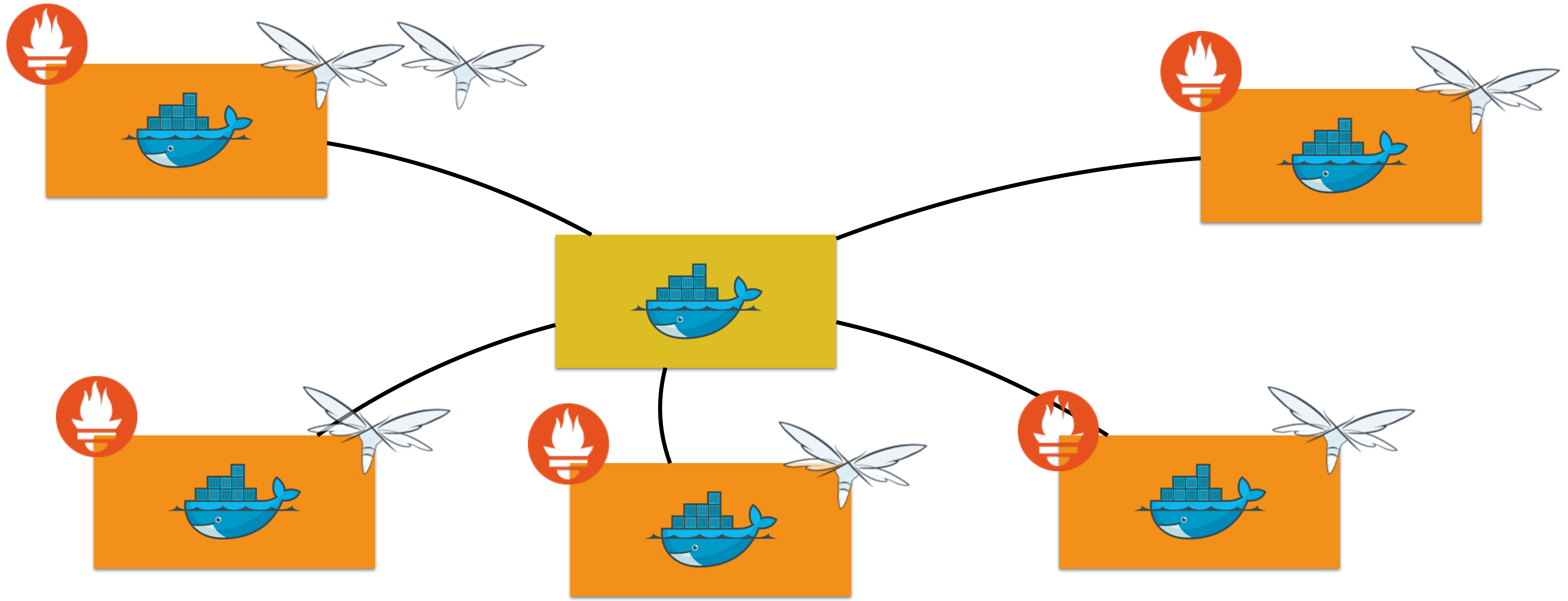


Swarm Mode: Scale



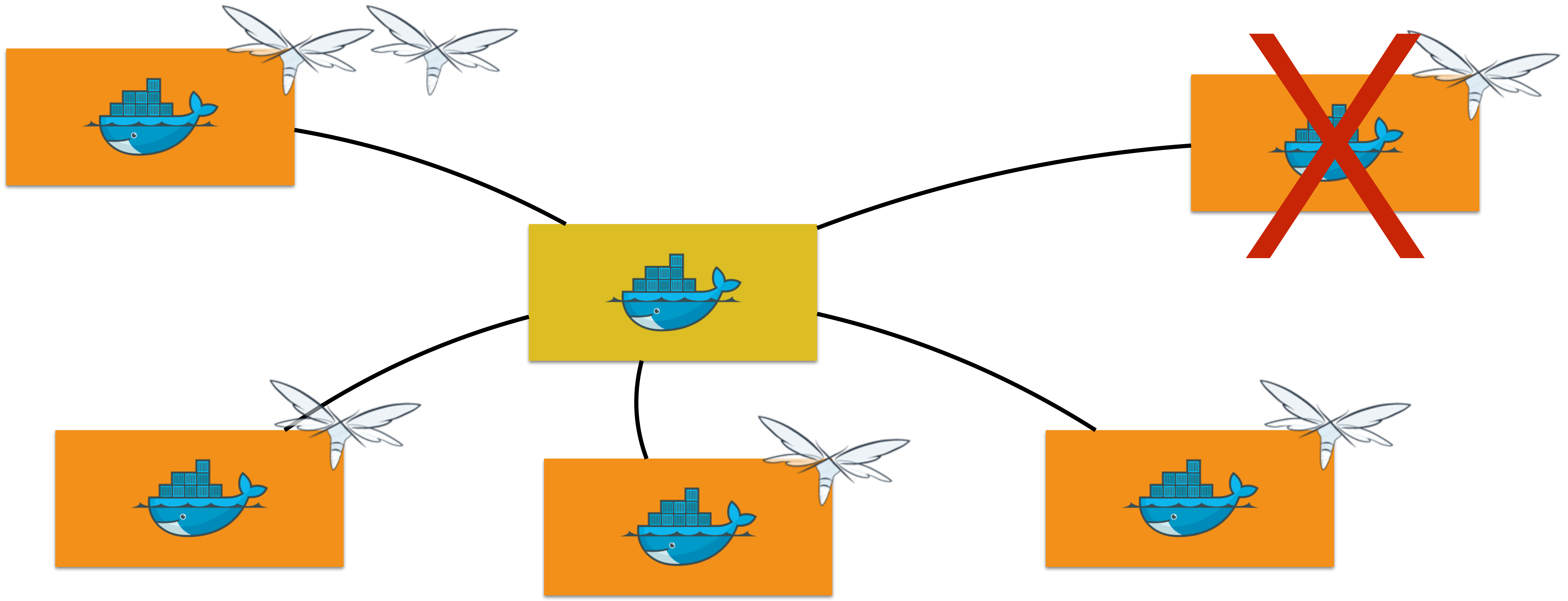
```
docker service scale web=6
```

Swarm Mode: Global Service



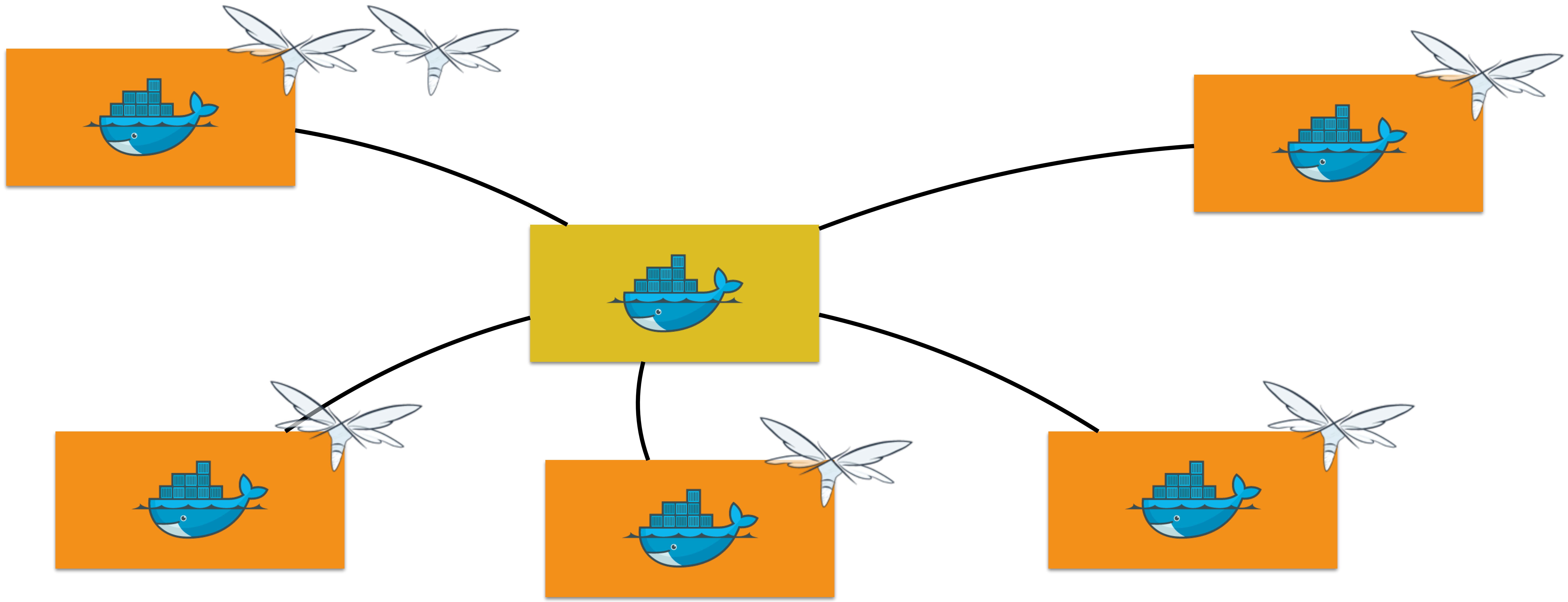
```
docker service create --mode=global --name=prom prom/prometheus
```

Swarm Mode: Pause Node



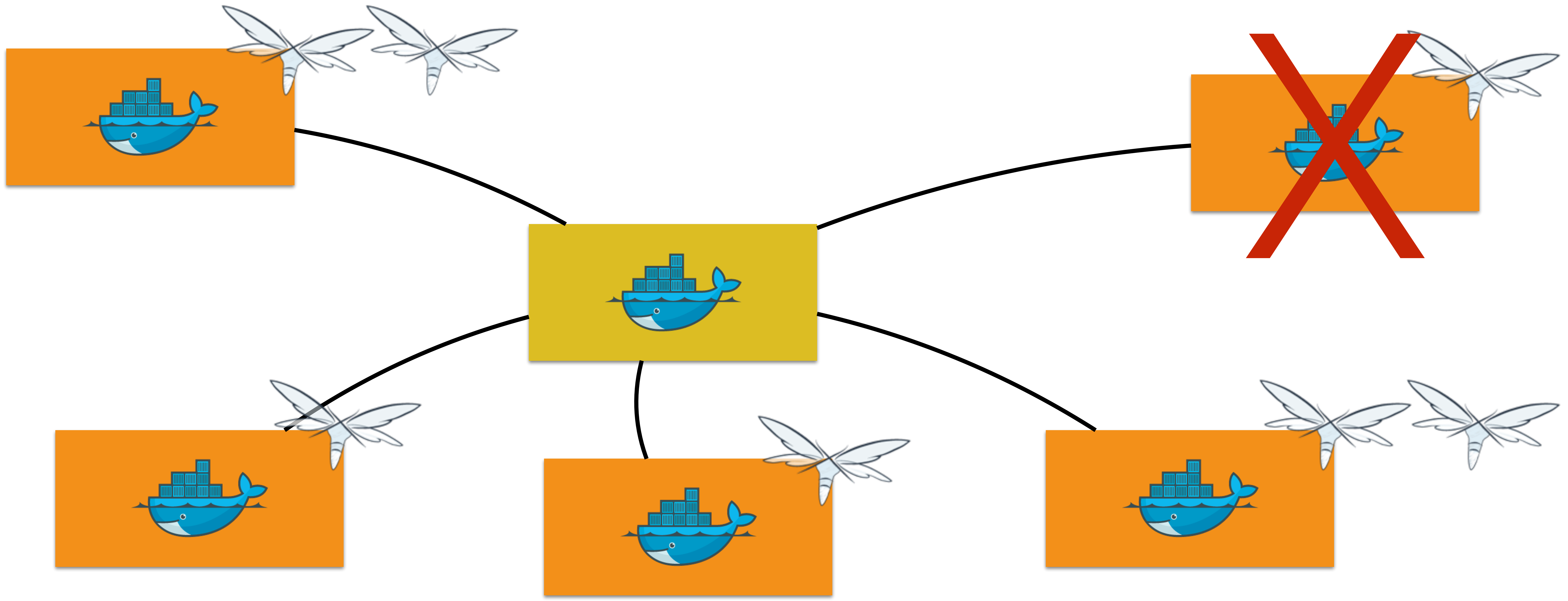
```
docker node update --availability pause <nodename>
```

Swarm Mode: Active Node



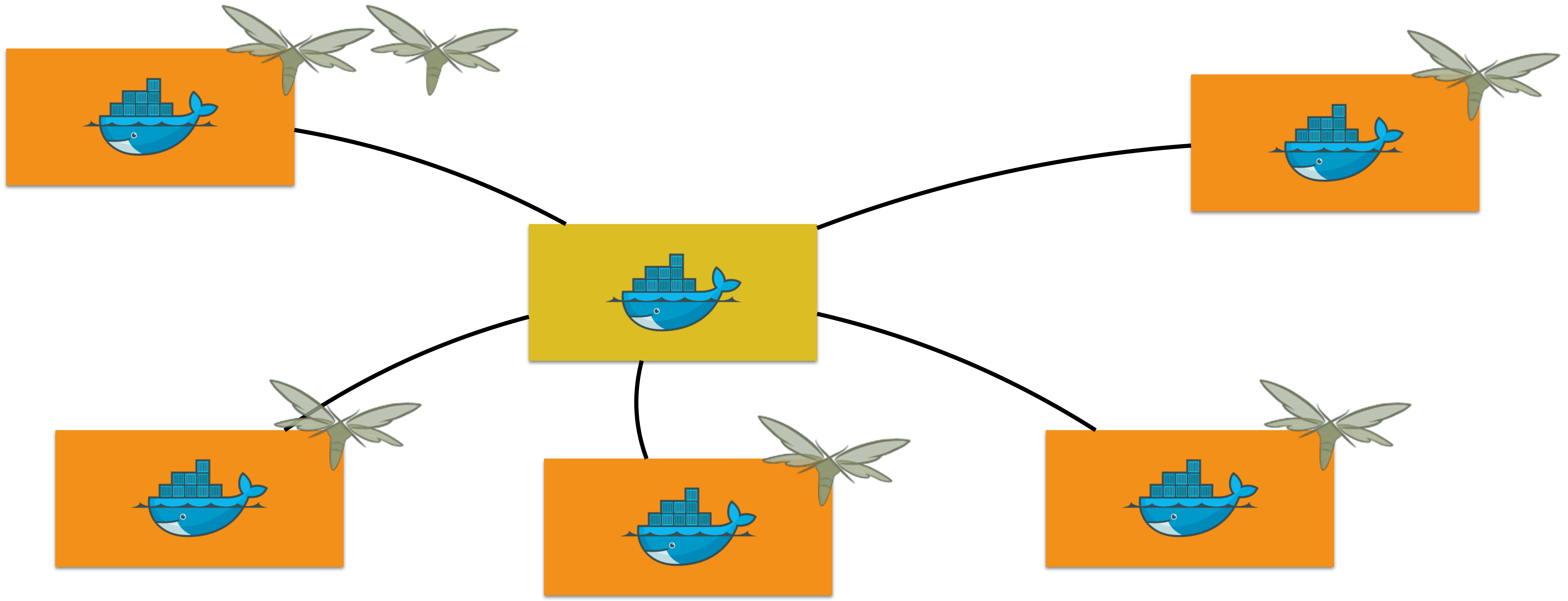
```
docker node update --availability active <nodename>
```

Swarm Mode: Drain Node



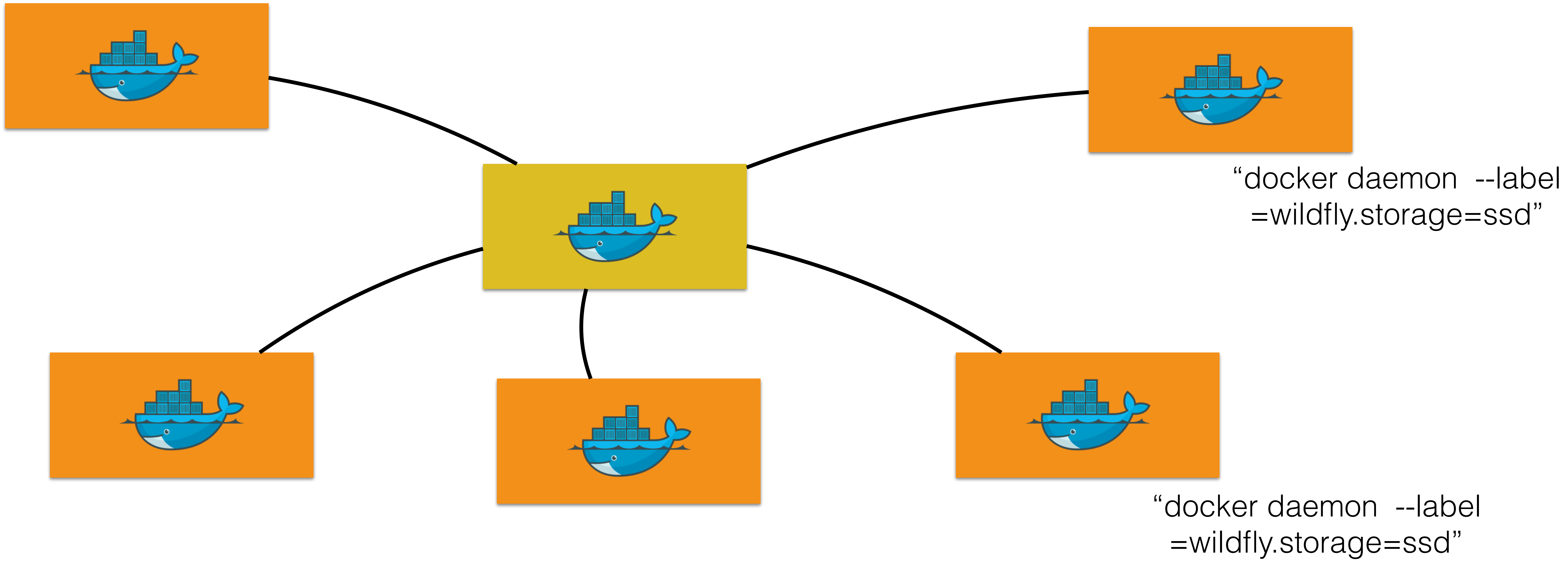
```
docker node update --availability drain <nodename>
```

Swarm Mode: Rolling Updates



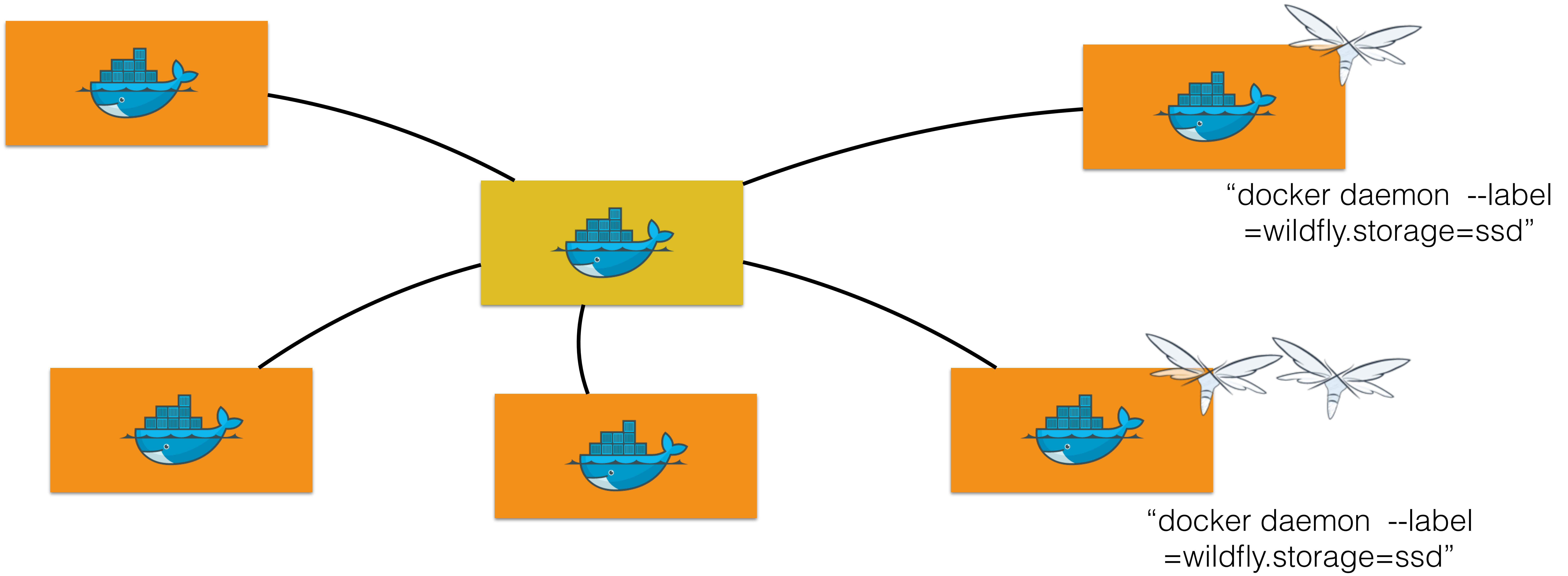
```
docker service update web --image wildfly:2 --update-parallelism
2 --update-delay 10s
```

Swarm Mode: Label


“docker daemon --label =wildfly.storage=ssd”
“docker daemon --label =wildfly.storage=ssd”

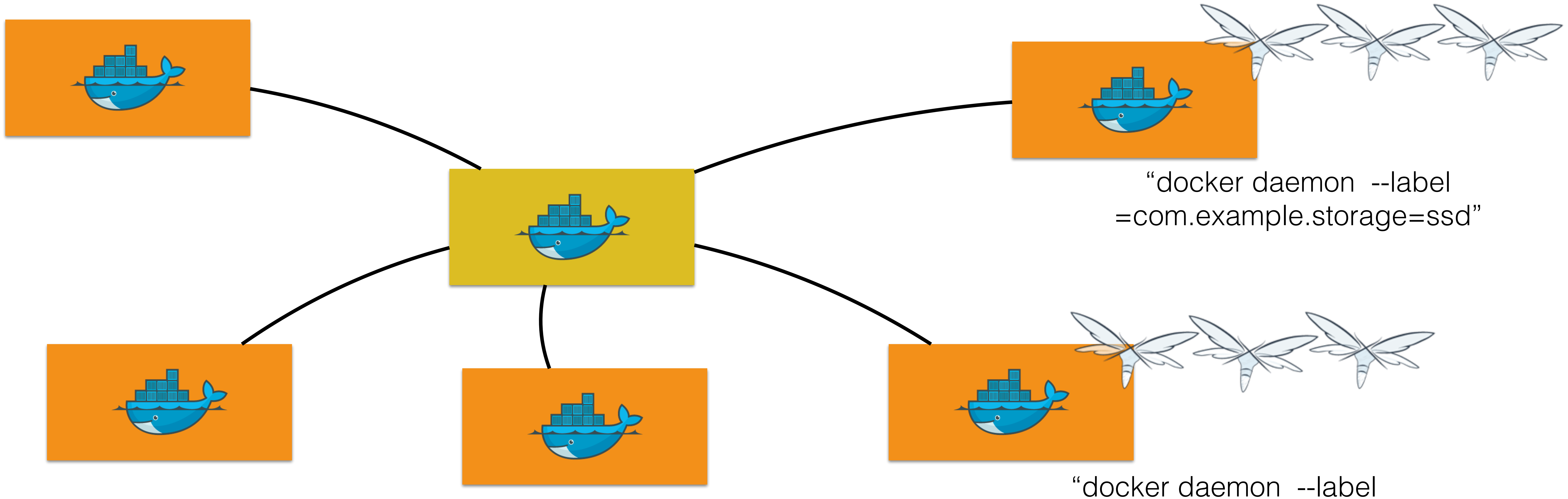
```
DOCKER_OPTS="--label=wildfly.storage=ssd"
```

Swarm Mode: Constraints


"docker daemon --label =wildfly.storage=ssd"
"docker daemon --label =wildfly.storage=ssd"

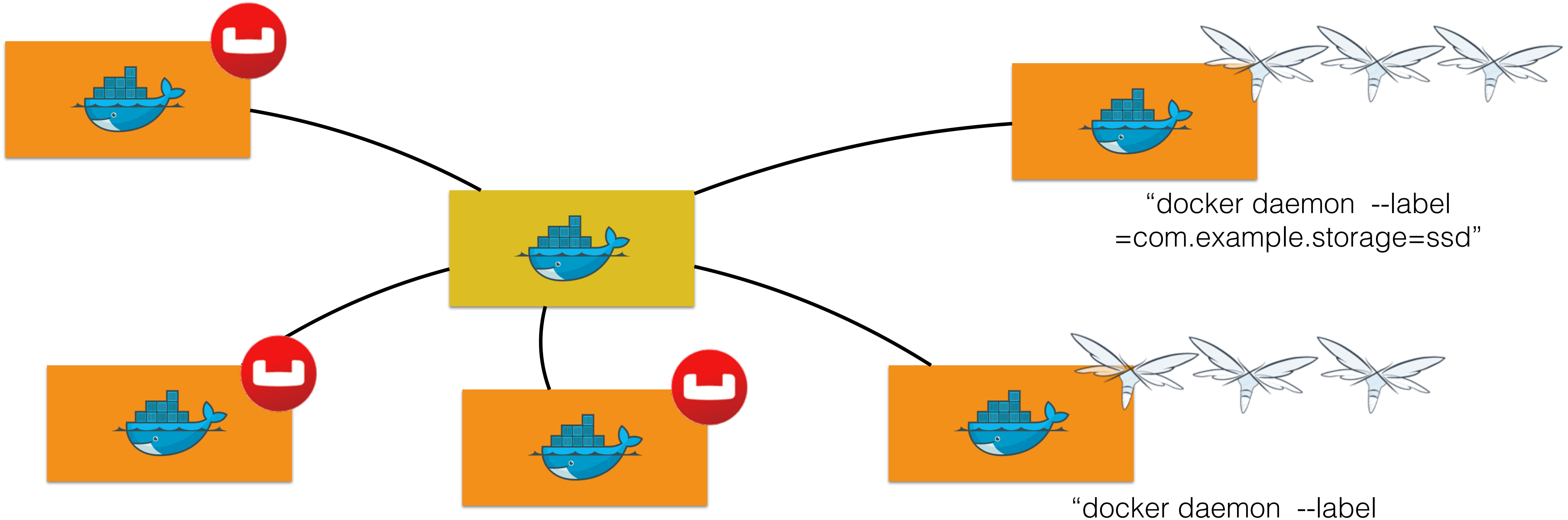
```
docker service create --replicas=3 --name=web --constraint
engine.labels.wildfly.storage==ssd jboss/wildfly
```

Swarm Mode: Constraints


“docker daemon --label =com.example.storage=ssd”
“docker daemon --label =com.example.storage=ssd”

```
docker service scale web=6
```

Swarm Mode: Constraints


“docker daemon --label =com.example.storage=ssd”
“docker daemon --label =com.example.storage=ssd”

```
docker service create --replicas=3 --name=db couchbase
```

Scheduling Backends using Filters

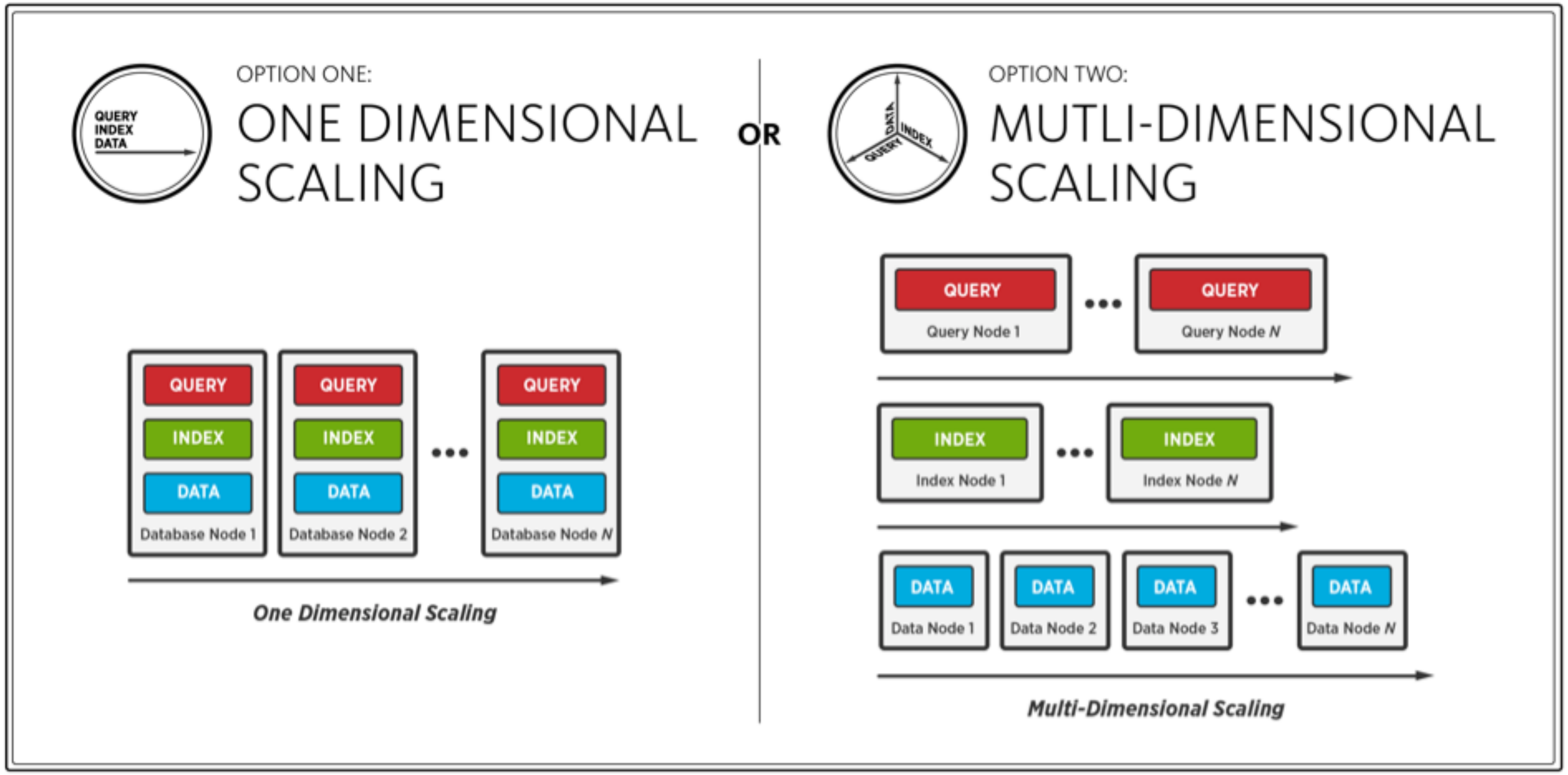
- **Label**: Metadata attached to Docker Daemon
- **Filters**: Used by Docker Swarm scheduler to create and run container

Node		
Constraint	Default or custom tags	`node`, `operatinsystem`, `kernelversion`, …
Health	Schedule containers on healthy nodes only	
Container Slots	Maximum number of containers on a node	`--labels containerslots=3`

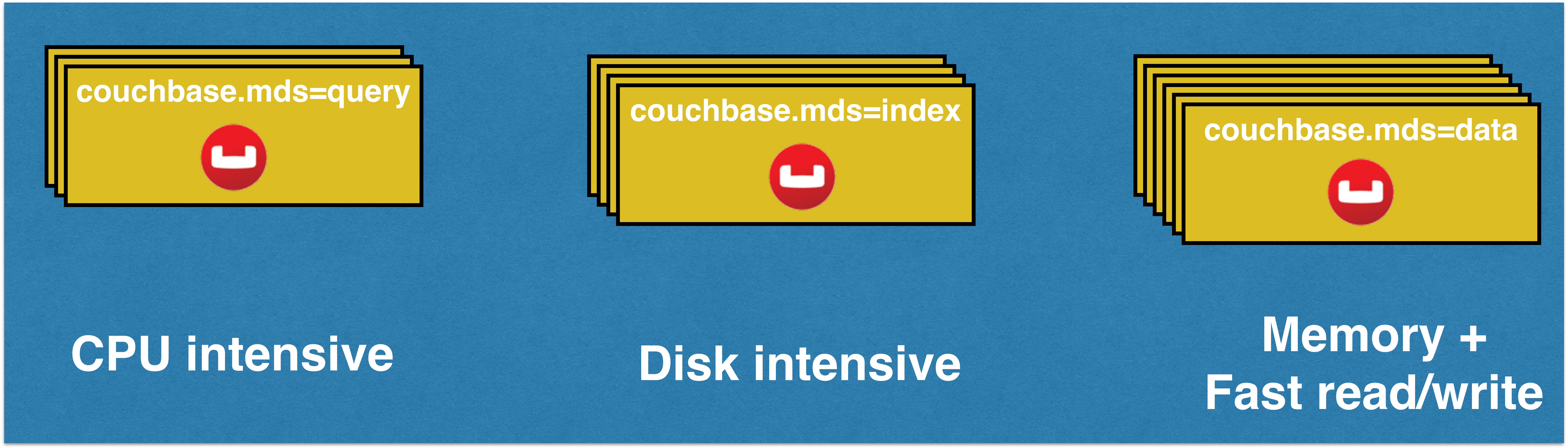
Container		
Affinity	“Attraction” between containers	`-e affinity:container=<name>/<id>`, `image`, …
Dependency	Dependent containers on same node	`--volumes-from=<id>`, `--net=container:<id>`, …
Port	Port availability on the host	`-p <host>:<container>`

Couchbase Multi Dimensional Scaling

Only Couchbase gives customers two options for scaling: Standard One-Dimensional Scaling and New Multi-Dimensional Scaling.

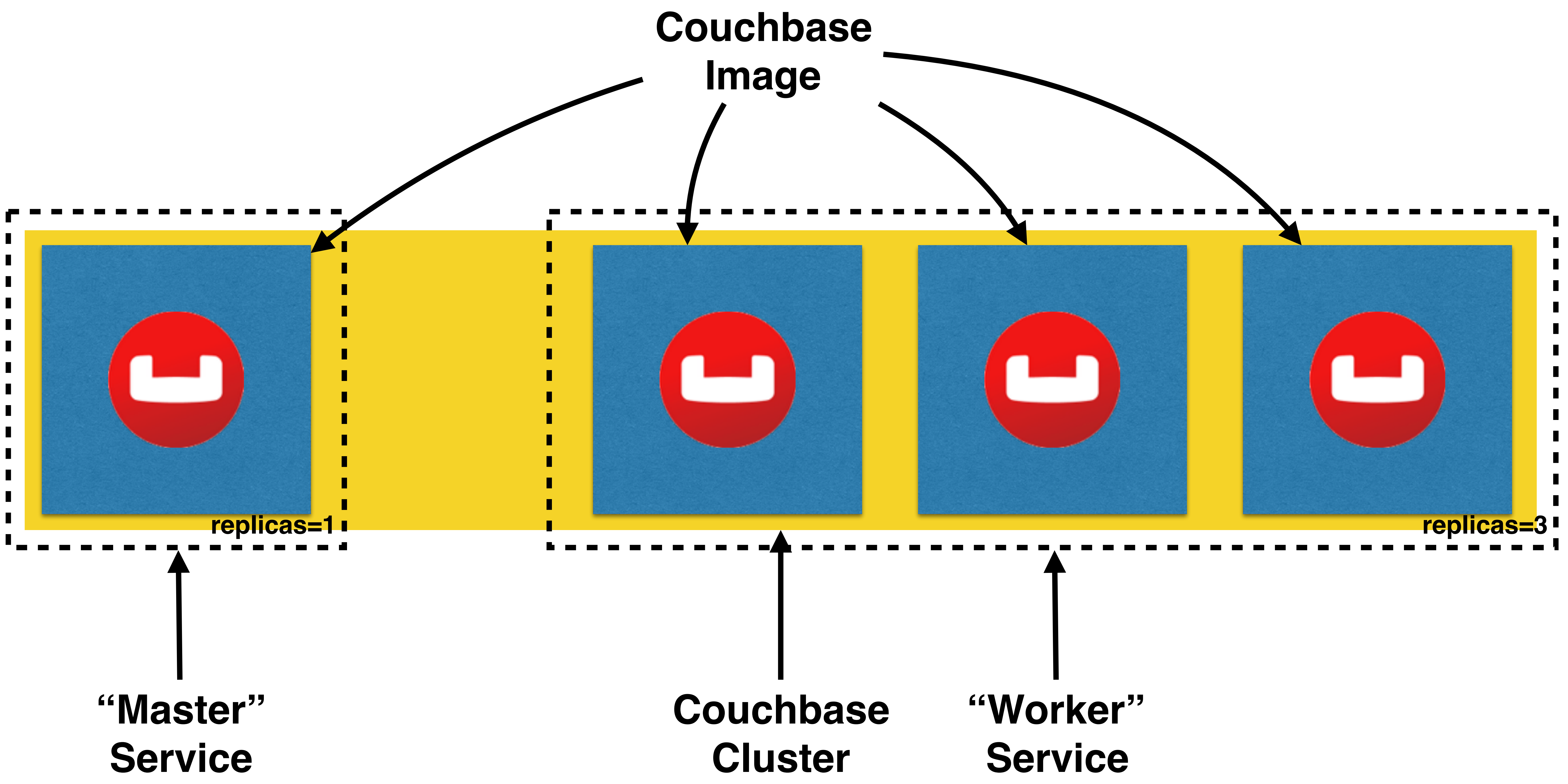


Optimal Utilization of Resources



- **Attach labels**: `DOCKER_OPTS="--label.couchbase.mds=data"`
- **Run Containers**: `docker service create --constraint engine.labels.couchbase.mds==index couchbase`

Couchbase Cluster using Docker Services

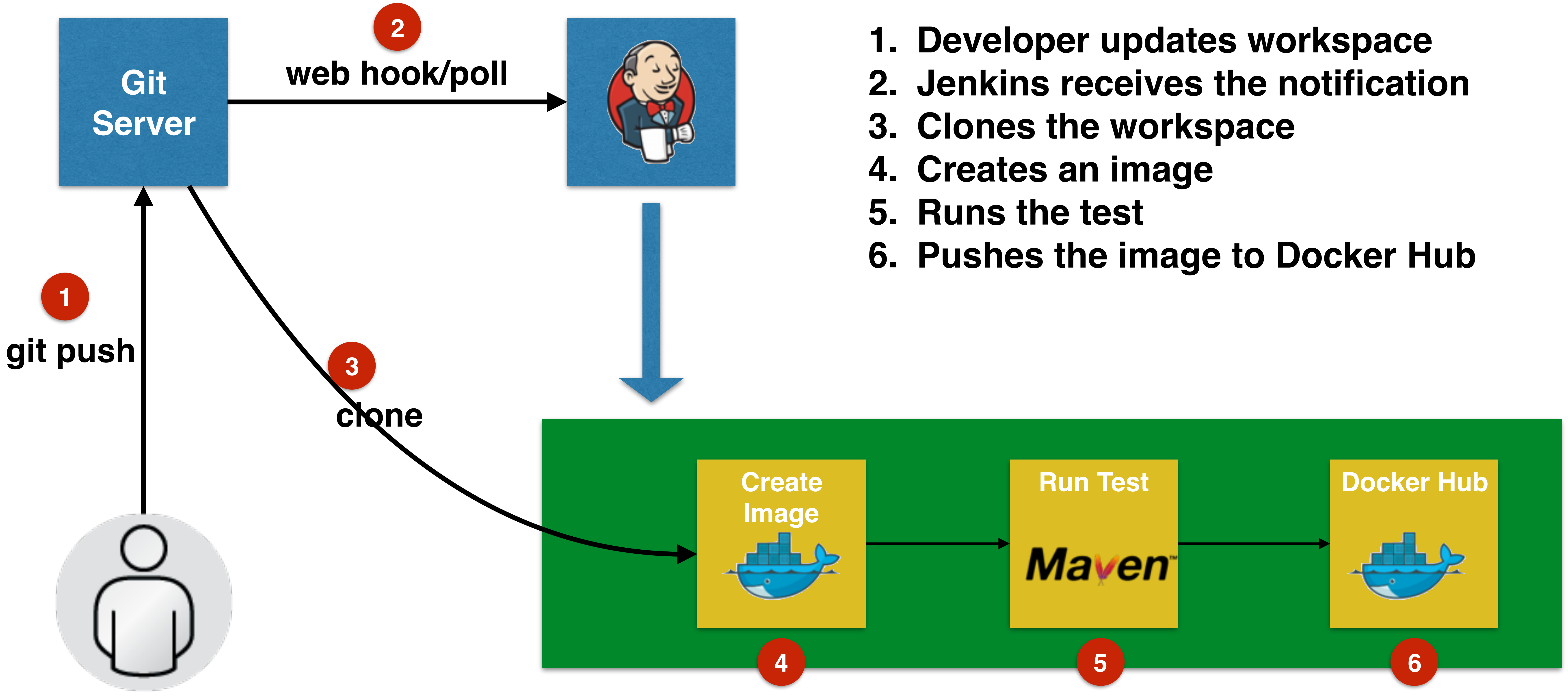


http://blog.couchbase.com/2016/september/docker-service-swarm-mode-couchbase-cluster

Monitoring Docker Containers

- `docker stats` command
  - LogEntries
- Service logs: `docker service logs <service>`
- Prometheus endpoint - New in Docker 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

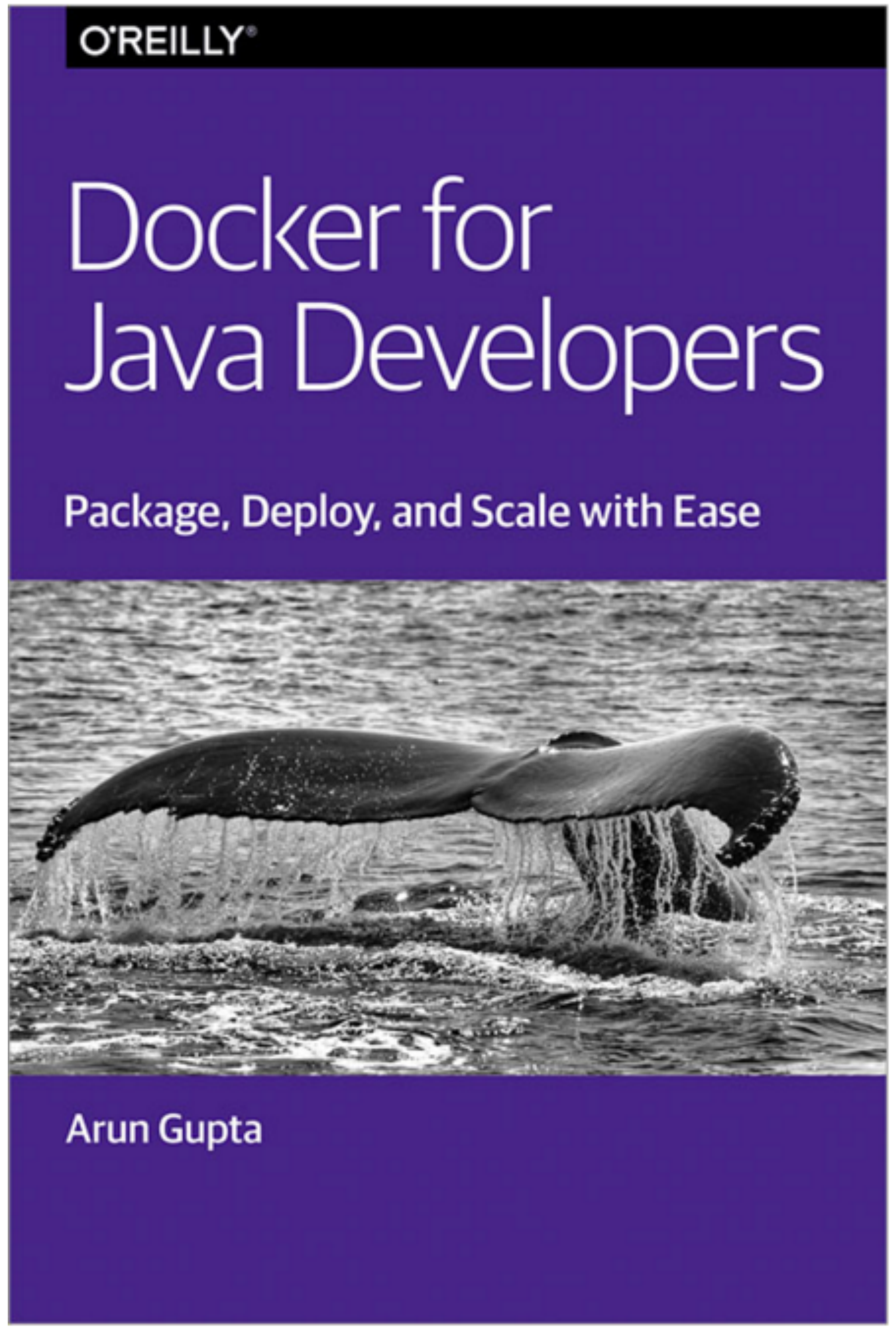
Docker Support in Java IDEs

NetBeans IDE 8.1

eclipse

JET BRAINS

https://github.com/arun-gupta/docker-java-ides bit.ly/dockerjava



O'REILLY

Docker for Java Developers

Package, Deploy, and Scale with Ease

Arun Gupta

bit.ly/kubejava

ONLINE COURSE

KUBERNETES FOR JAVA DEVELOPERS

September 23 | 10AM-12PM

Arun Gupta

References

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