Capgemini

Package your application with Docker

Craftsmanship Community
Wroclaw, 27.04.2018

Krzysztof Sobkowiak

Disclaimer

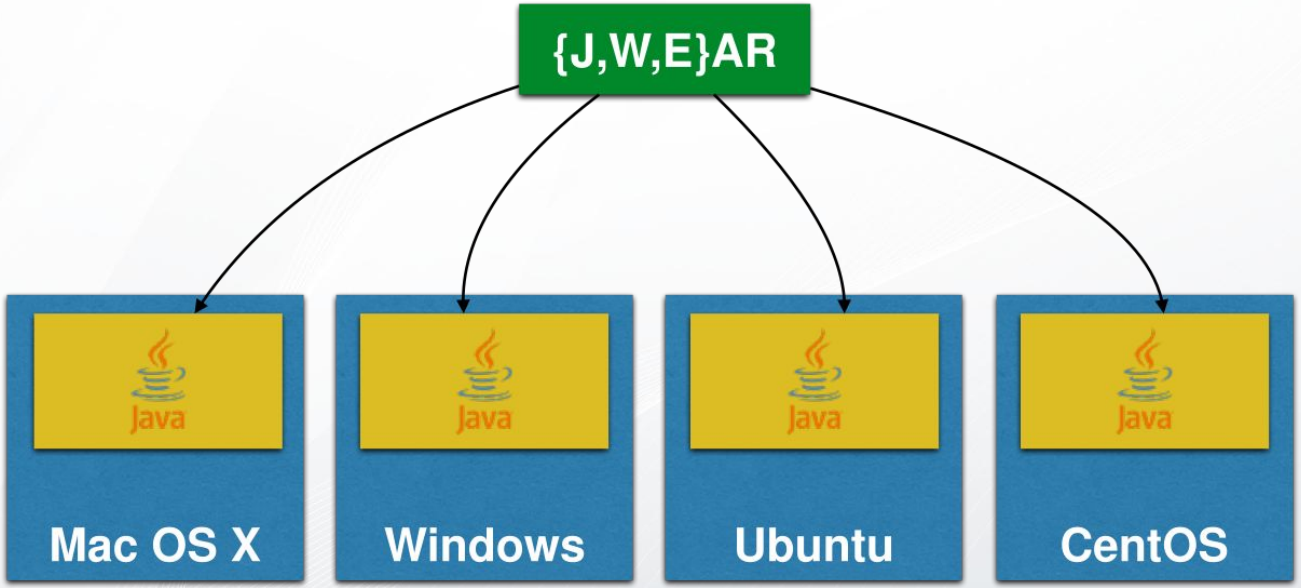
Views in this presentation are my personal views and do not necessarily reflect the views of Capgemini.

Introduction

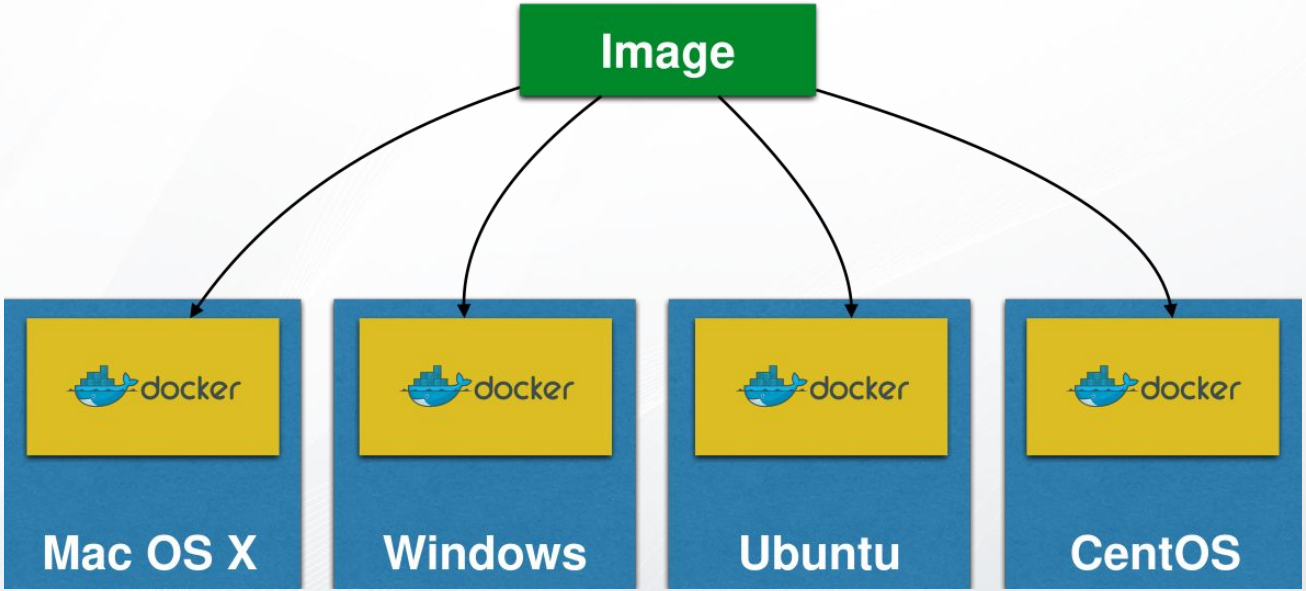
Creating business value through software is about speed, safety, iteration, and continuous improvement

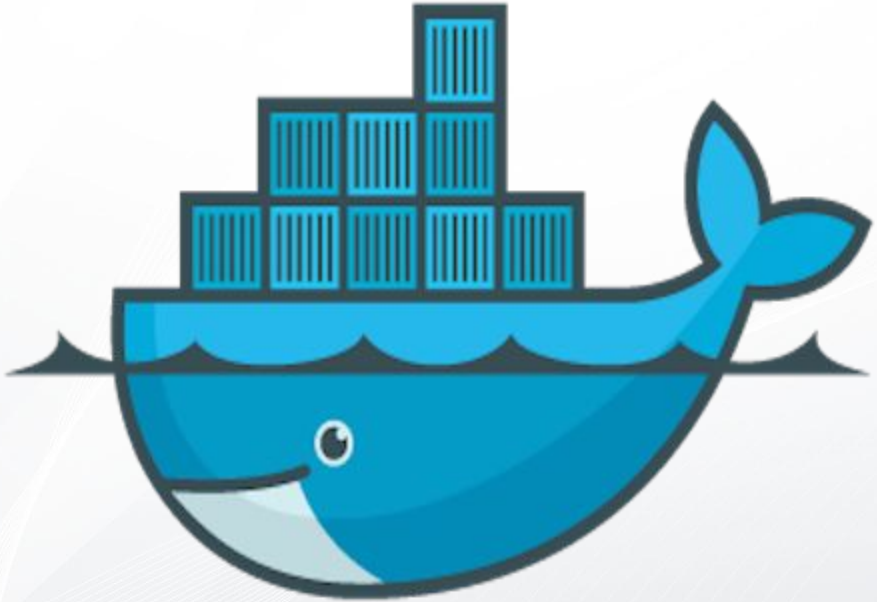
Capgemini

WORA - Write Once Run Anywhere



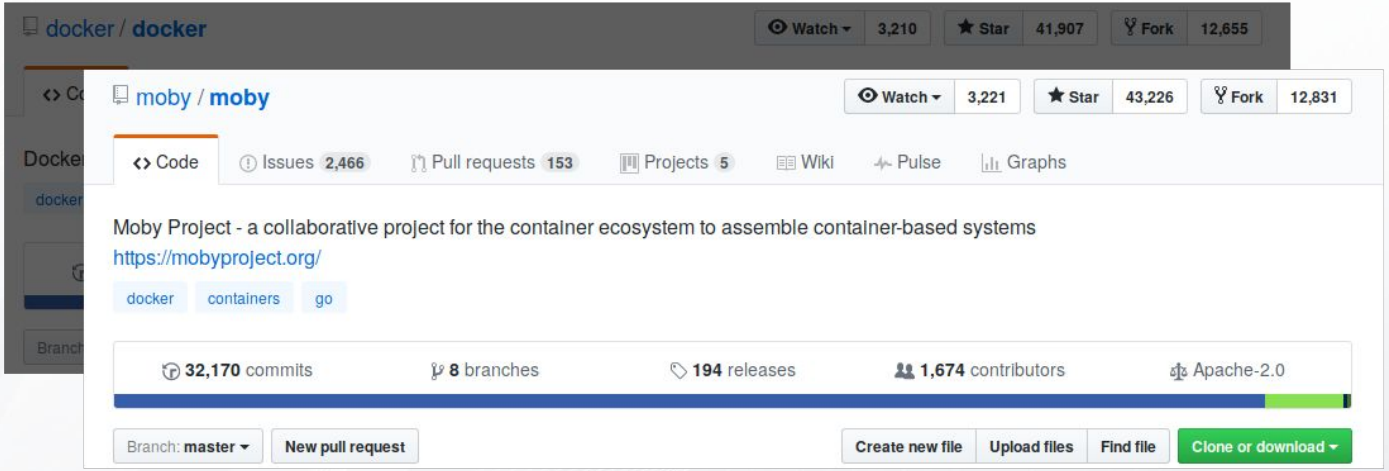
PODA - Package Once Deploy Anywhere





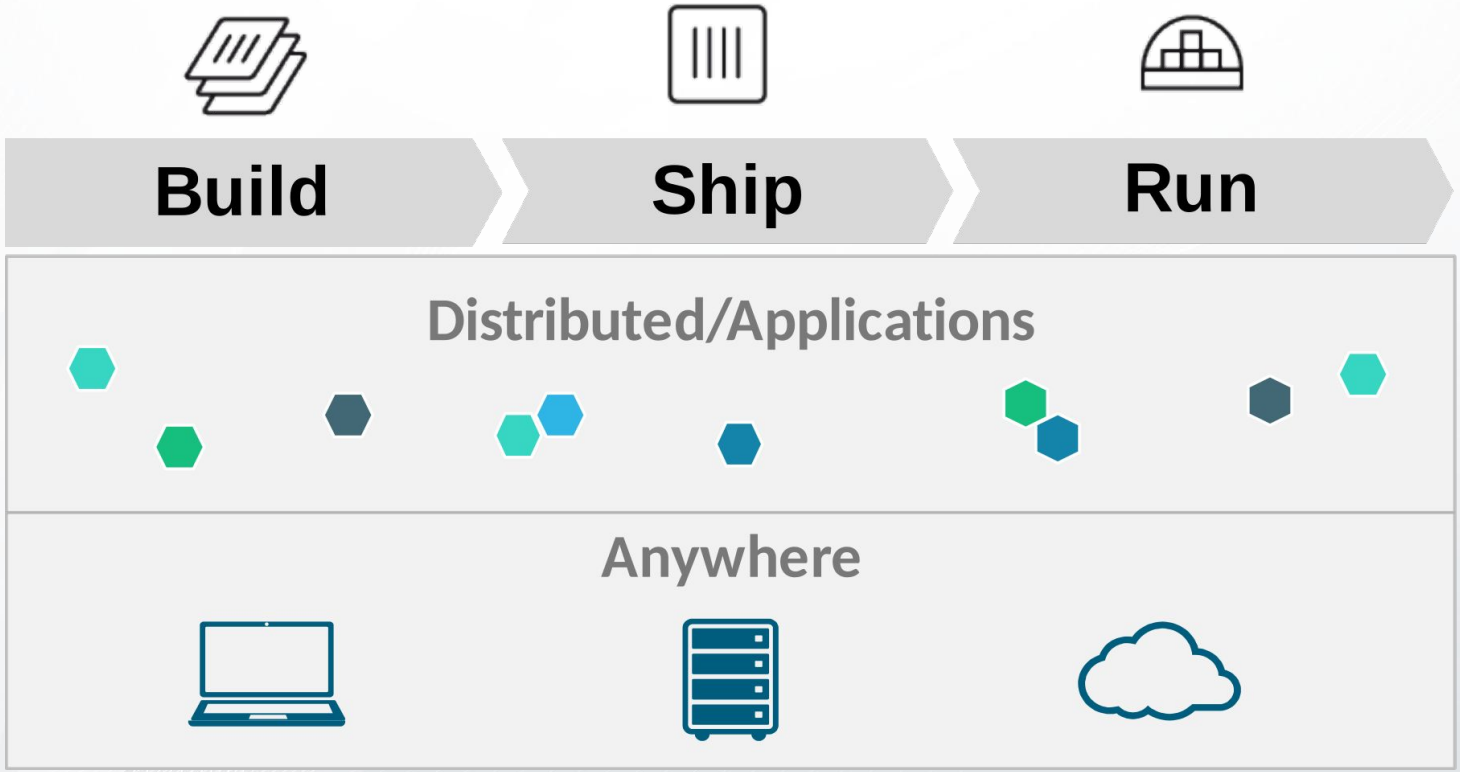
What is Docker?

- Open source project and company

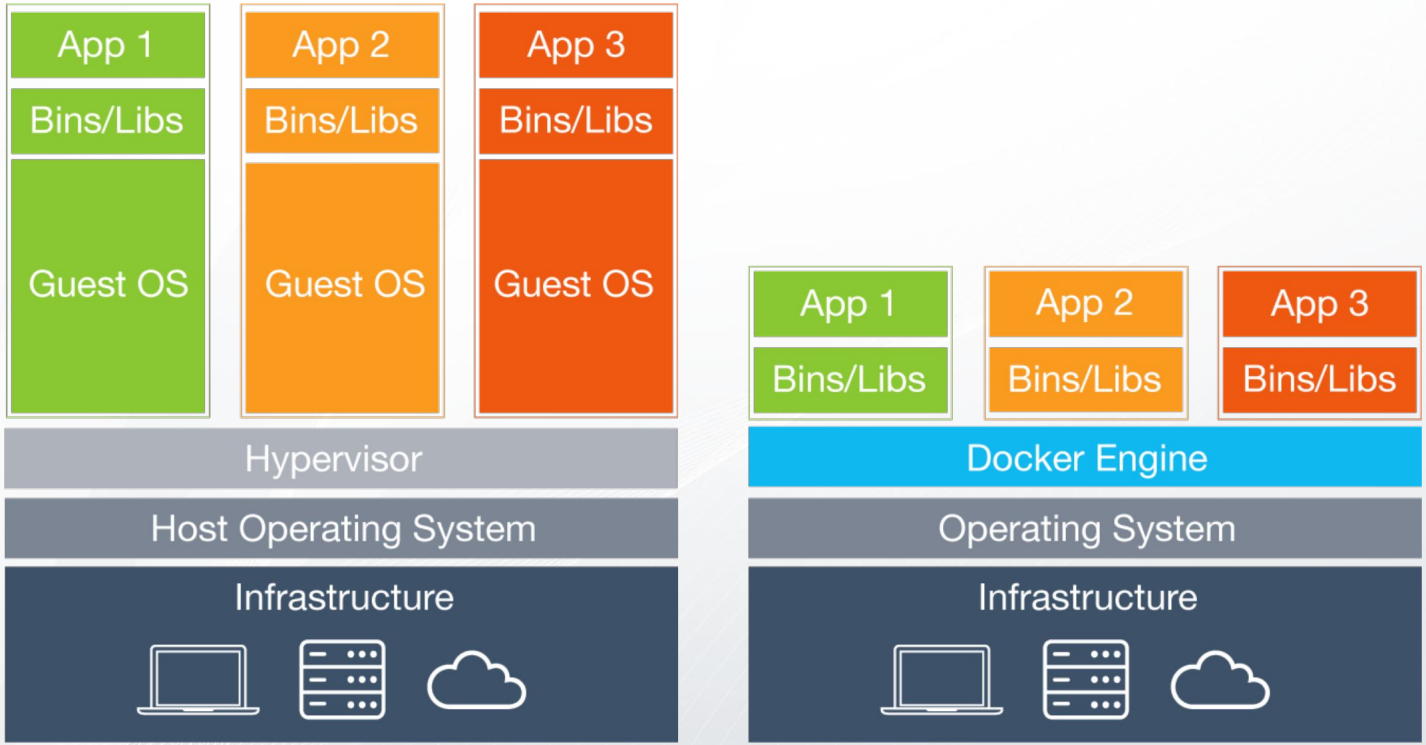


- Used to create containers for software applications
- Package Once Deploy Anywhere (PODA)

Docker Mission

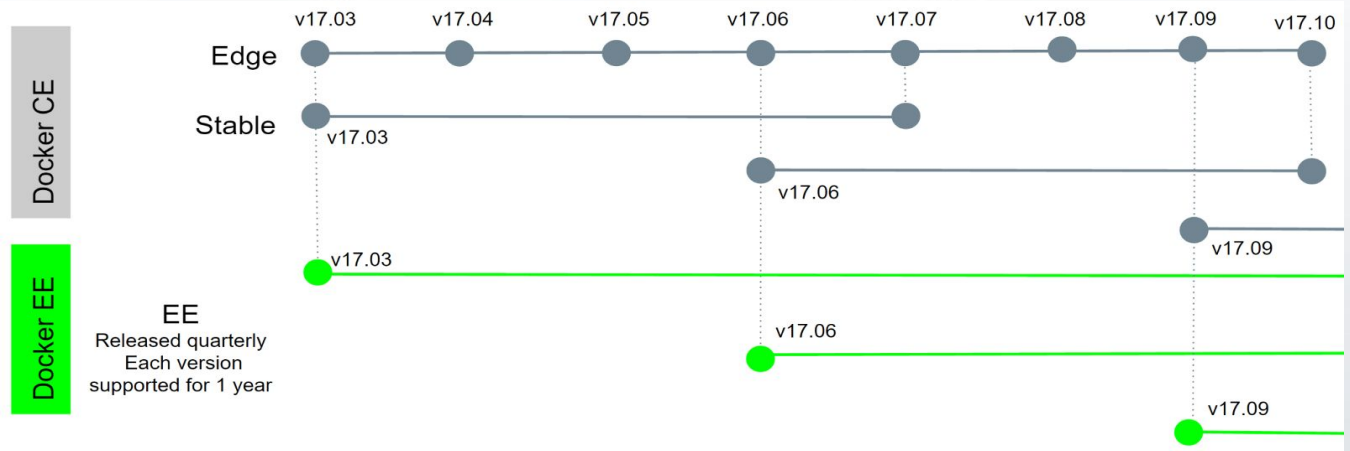


VMs vs Containers



Install Docker

- Available in 2 editions: Community Edition (CE) and Enterprise Edition (EE).
- Docker for Mac/Windows (Desktop) and Linux (Server)
- Time-based release schedule
  - CE: Monthly edge and quarterly stable release
  - EE: Released quarterly, each version supported for 1 year



https://docs.docker.com/install

Docker Machine

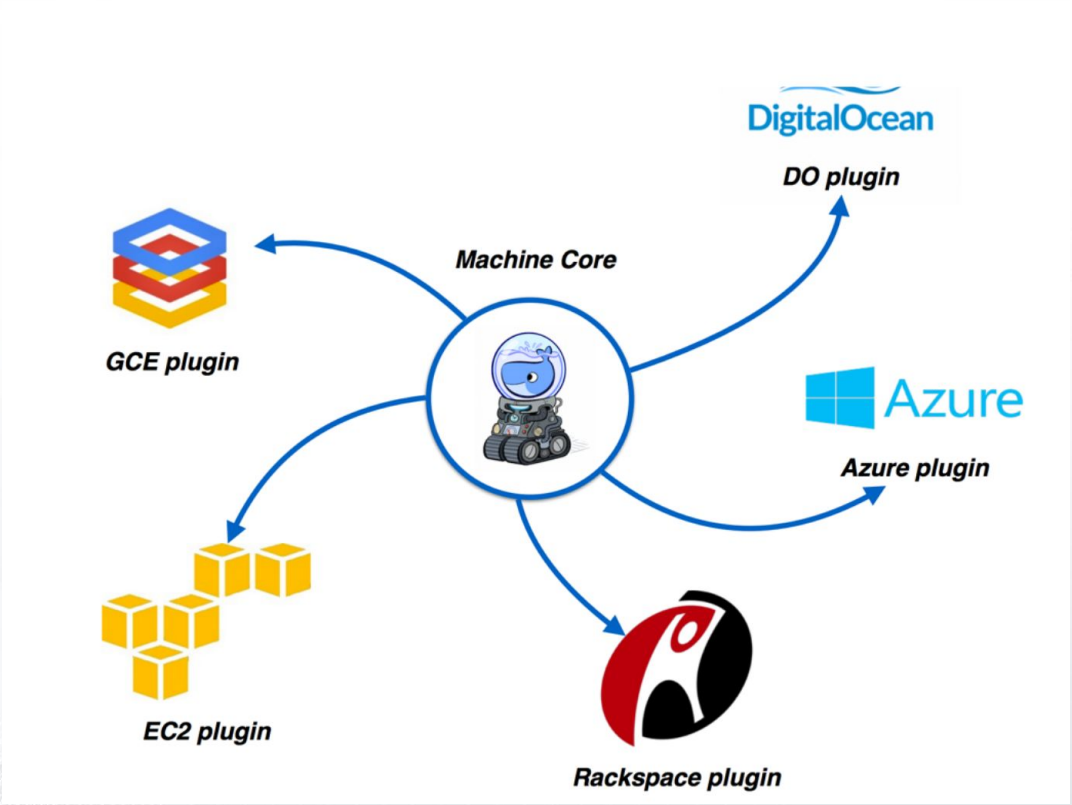
- Create Docker Host on computer or cloud provider

```
docker-machine create --driver=virtualbox myhost
```

  - Configure Docker client to talk to host
  - Create and pull images
  - Start, stop, restart containers
  - Upgrade Docker

https://docs.docker.com/machine/overview/

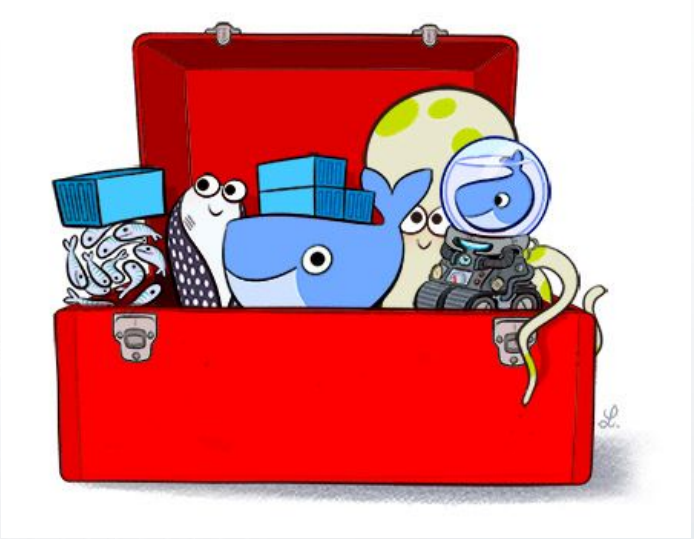
Docker Machine Providers



Docker Toolbox

DEPRECATED

- Docker Client
- Docker Machine
- Docker Compose
- Docker Kitematic
- Boot2Docker ISO
- Virtualbox



https://docs.docker.com/toolbox/overview/

Capgemini

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
- Better networking and filesystem mounting/notification
- Requires Yosemite 10.10+ or Windows 10 64-bit

https://docs.docker.com/docker-for-mac
https://docs.docker.com/docker-for-windows

Capgemini

amazon web services

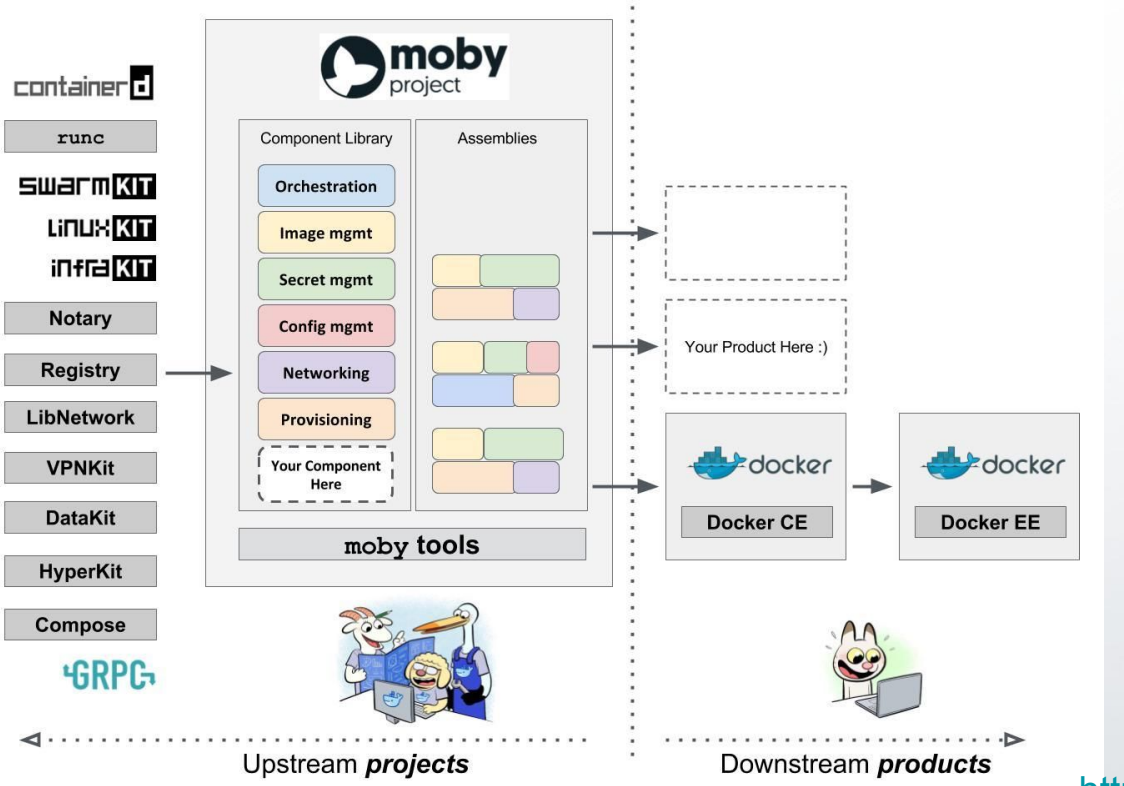
Microsoft Azure

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

https://docs.docker.com/docker-for-aws/
https://docs.docker.com/docker-for-azure/

Meet the Moby

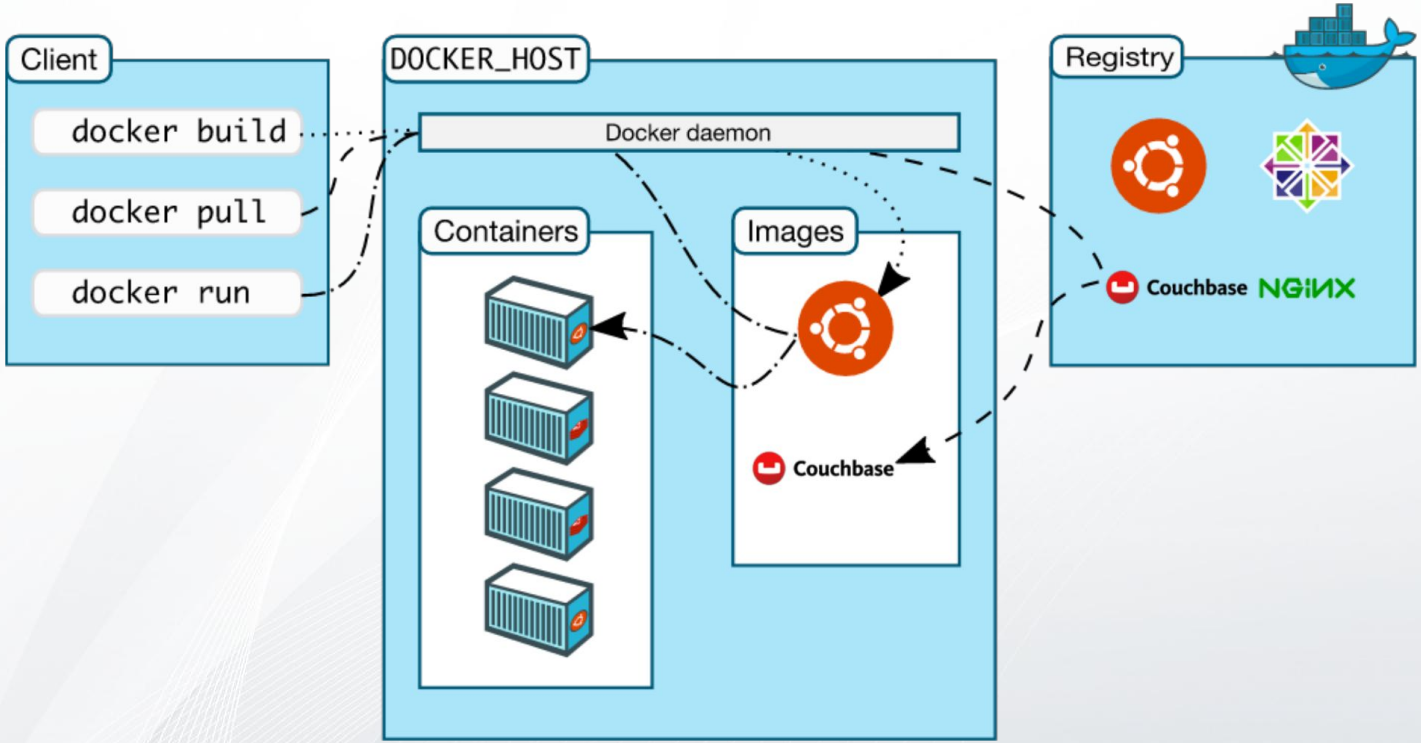


https://mobyproject.org/

Capgemini

Docker Workflow

Docker Workflow



Capgemini

Build

- Image defined in text-based Dockerfile
- List of commands to build the image

```
FROM ubuntu

ENTRYPOINT ["echo", "Hello world!!!"]
```

- Docker commands to build the image

```
docker build -t org/hello .
```

Dockerfile

```
FROM ubuntu
MAINTAINER Kimbro Staken

RUN apt-get install -y software-properties-common python
RUN add-apt-repository ppa:chris-lea/node.js
RUN echo "deb http://us.archive.ubuntu.com/ubuntu/ precise universe" >> /etc/apt/sources.list
RUN apt-get update
RUN apt-get install -y nodejs
#RUN apt-get install -y nodejs=0.6.12~dfsg1-1ubuntu1
RUN mkdir /var/www

ADD app.js /var/www/app.js

CMD ["/usr/bin/node", "/var/www/app.js"]
```

https://docs.docker.com/engine/reference/builder/

Union File System

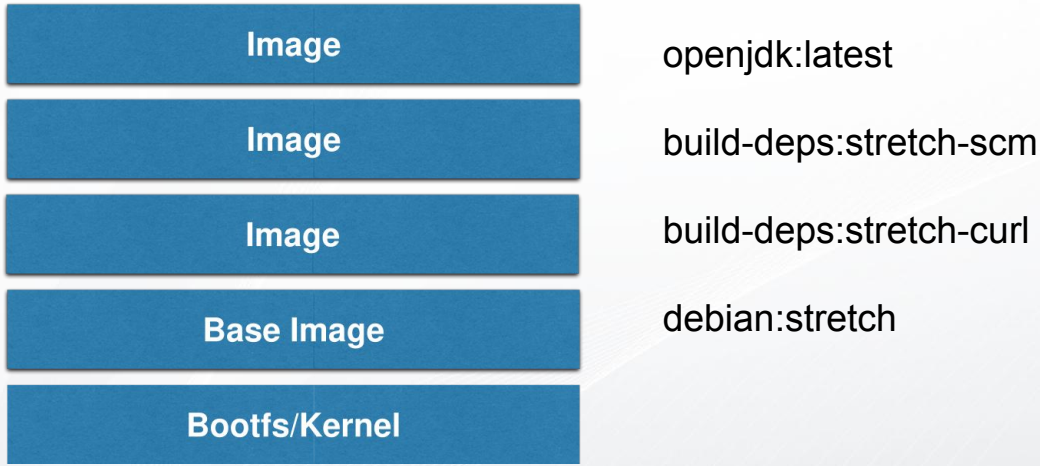


Image Layers - OpenJDK

```
$ docker images openjdk
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
openjdk             latest              891c9734d5ab        5 weeks ago         726MB

$ docker history openjdk
IMAGE               CREATED             CREATED BY                                      SIZE
      COMMENT
891c9734d5ab        5 weeks ago         /bin/sh -c /var/lib/dpkg/info/ca-certificate…   394kB
<missing>           5 weeks ago         /bin/sh -c set -ex;   if [ ! -d /usr/share/m…   450MB
<missing>           5 weeks ago         /bin/sh -c #(nop)  ENV CA_CERTIFICATES_JAVA_…   0B
<missing>           5 weeks ago         /bin/sh -c #(nop)  ENV JAVA_DEBIAN_VERSION=8…   0B
<missing>           5 weeks ago         /bin/sh -c #(nop)  ENV JAVA_VERSION=8u162       0B
<missing>           5 weeks ago         /bin/sh -c #(nop)  ENV JAVA_HOME=/docker-jav…   0B
<missing>           5 weeks ago         /bin/sh -c ln -svT "/usr/lib/jvm/java-8-open…   33B
<missing>           5 weeks ago         /bin/sh -c {   echo '#!/bin/sh';   echo 'set…   87B
<missing>           5 weeks ago         /bin/sh -c #(nop)  ENV LANG=C.UTF-8             0B
<missing>           5 weeks ago         /bin/sh -c apt-get update && apt-get install…   2.21MB
<missing>           5 weeks ago         /bin/sh -c apt-get update && apt-get install…   142MB
<missing>           5 weeks ago         /bin/sh -c set -ex;  if ! command -v gpg > /…   7.8MB
<missing>           5 weeks ago         /bin/sh -c apt-get update && apt-get install…   23.8MB
<missing>           5 weeks ago         /bin/sh -c #(nop)  CMD ["bash"]                 0B
<missing>           5 weeks ago         /bin/sh -c #(nop) ADD file:b380df301ccb5ca09…   100MB
```

Capgemini

Ship

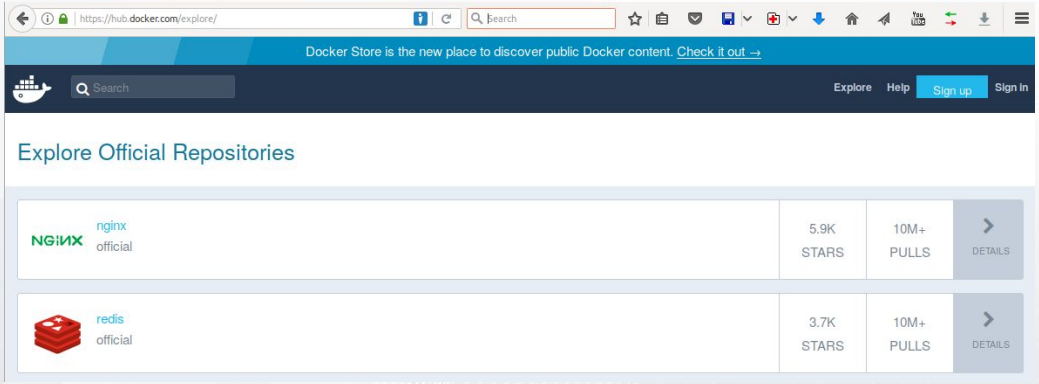
- Store and distribute Docker images with Docker Registry
  - Control where images are stored
  - Own image distribution pipeline
  - Integrate image storage/distribution in dev workflow

- Docker commands to push or pull the image

```
docker push org/hello
docker pull org/hello
```

Docker registry

- Docker Hub
  - Free-to-use and hosted
- Docker Trusted Registry
  - Commercially supported
  - RBAC, LDAP/AD integration, updates, etc
  - Can be setup inside firewall

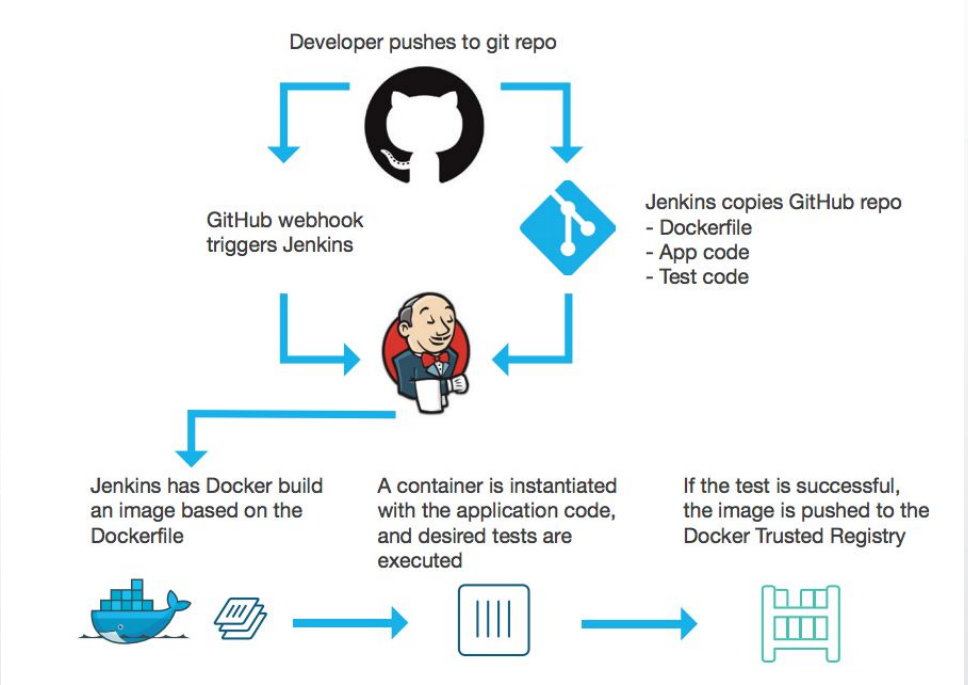


https://docs.docker.com/registry/

https://docs.docker.com/datacenter/dtr/2.3/guides/

Registry primary usage

- CI/CD with Docker
  - Centrally located base images
  - Store individual build images
  - Pull tested images to production



Run

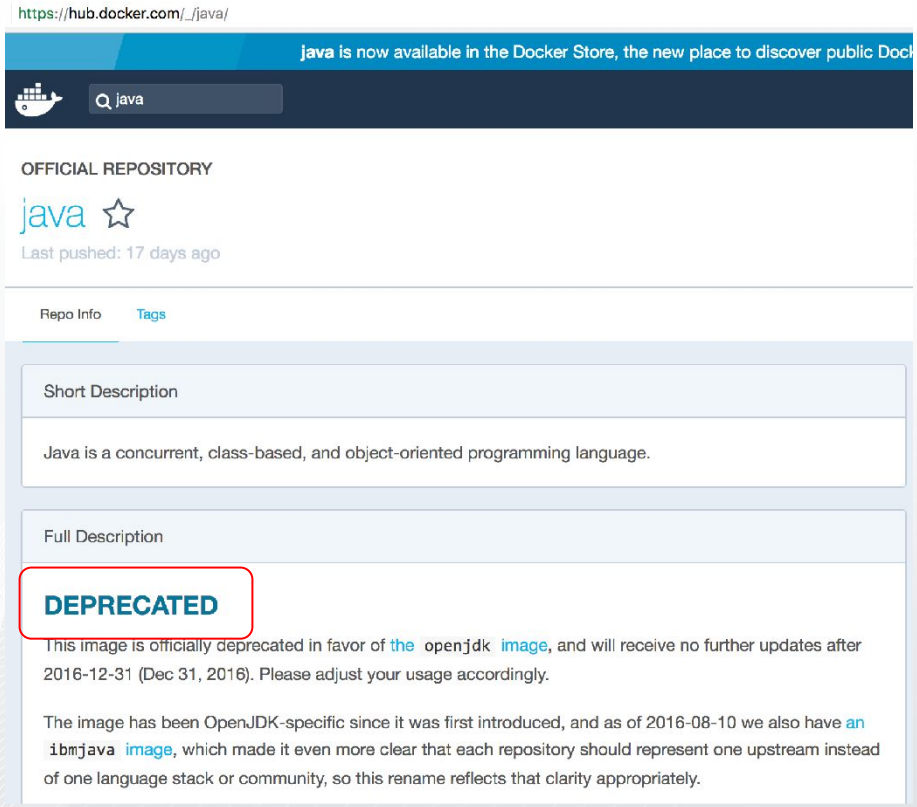
- Container built from the image
- Runtime representation of the image
- Self contained execution environment

```
docker run org/hello
```

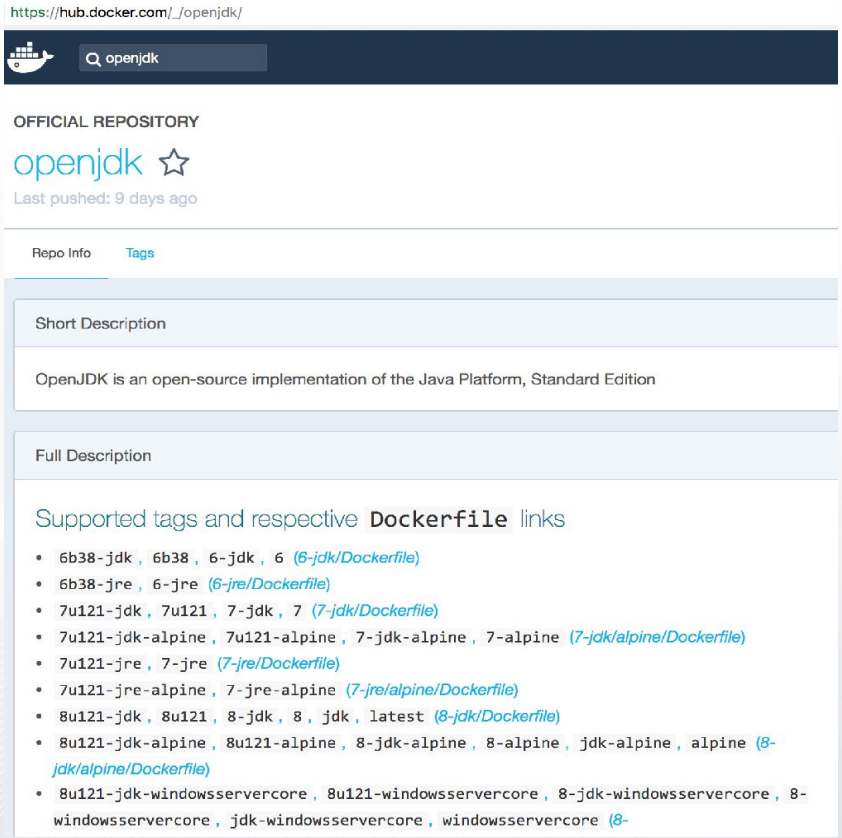
Capgemini

Docker for Java developers

Java Base Image



Java Base Image



First Java application

```
FROM openjdk:latest

ADD target/hello.jar .

EXPOSE 8080
ENTRYPOINT ["/usr/bin/java", "-jar", "hello.jar"]
```

Maven Plugin

- Plugin

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

- Goals: **`docker:X`**, X= stop, build, push, ...

```
<plugin>
  <groupId>io.fabric8</groupId>
  <artifactId>docker-maven-plugin</artifactId>
  <version>0.21.0</version>
  <configuration>
    <images>
      <image>
        <alias>itemservice</alias>
        <name>sobkowiak/itemservice:latest</name>
        <build>
          <from>openjdk:8-jre</from>
          <assembly>
            <descriptorRef>artifact</descriptorRef>
          </assembly>
          <cmd>java -jar maven/${project.artifactId}.jar
          </cmd>
        </build>
        <run>
          <network>
            <mode>custom</mode>
            <name>itemservice</name>
            <alias>itemservice</alias>
          </network>
        </run>
      </image>
    </images>
  </configuration>
</plugin>
```

https://dmp.fabric8.io/

Capgemini

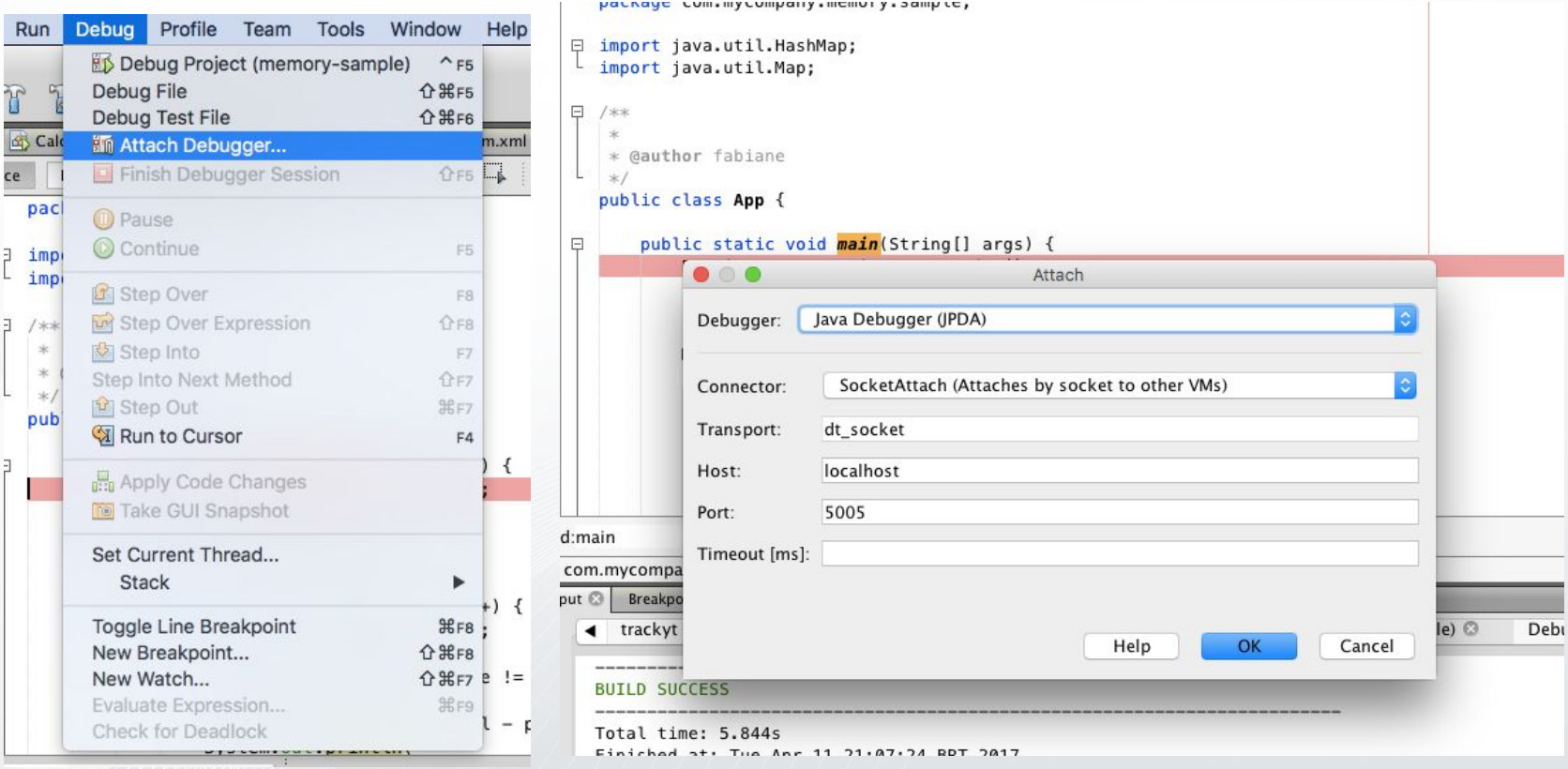
Docker Support in Java IDEs

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

Running in debug mode

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

Attaching the IDE



Multi-Container Application

Capgemini

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

https://docs.docker.com/compose/overview/

Capgemini

Multi-container on single host

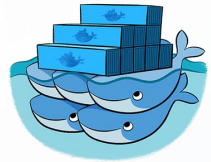
```yaml
version: "3"
services:
  itemsdb:
    image: postgres
    ports:
    - "5432:5432"
    environment:
    - POSTGRES_PASSWORD=items
    - POSTGRES_USER=items
    - POSTGRES_DB=items
  itemservice:
    image: sobkowiak/itemservice:latest
    ports:
    - "8090:8090"
    depends_on:
    - itemsdb
    environment:
    - POSTGRES_URL=itemsdb
```

```
docker-compose up -d
```

Capgemini

Multi-host using Swarm Mode

Capgemini

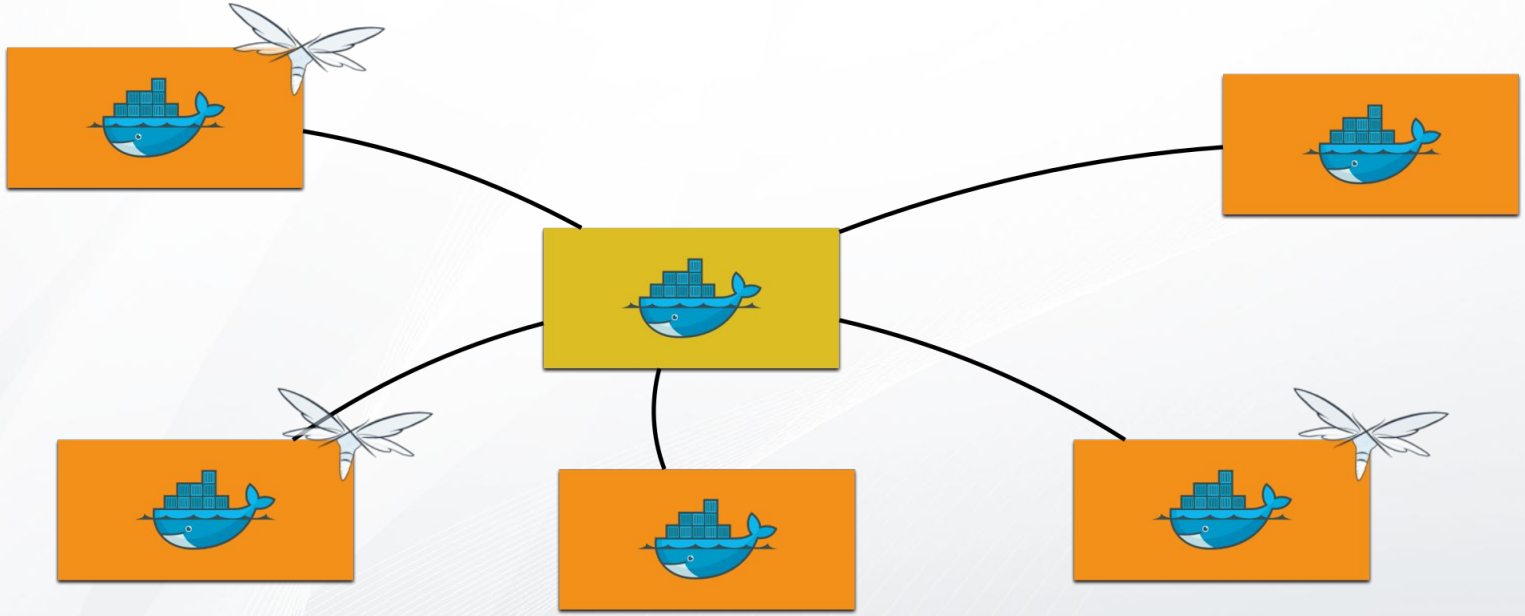


Swarm Mode

- 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

https://docs.docker.com/engine/swarm/

Swarm Mode - Replicated Service



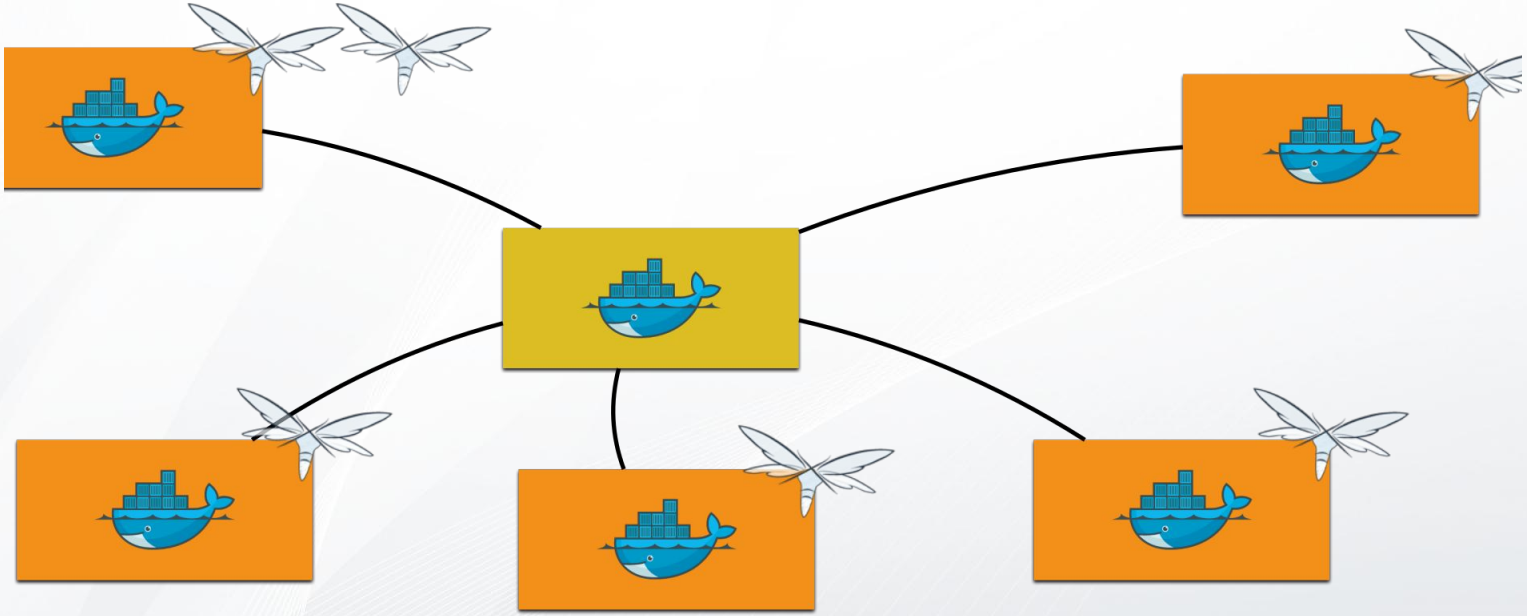
```
docker service create --replicas 3 --name itemservice sobkowiak/itemservice
```

Multi-container on multiple hosts

```yaml
version: "3"
services:
  itemsdb:
    image: postgres
    ports:
    - "5432:5432"
    environment:
    - POSTGRES_PASSWORD=items
    - POSTGRES_USER=items
    - POSTGRES_DB=items
  itemservice:
    image: sobkowiak/itemservice:latest
    ports:
    - "8090:8090"
    depends_on:
    - itemsdb
    environment:
    - POSTGRES_URL=itemsdb
```

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

Swarm Mode - Replicated Service

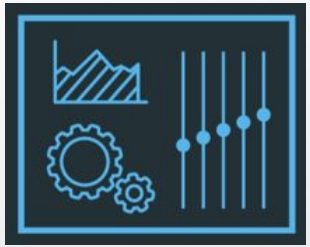


```
docker service scale itemservice=6
```

Monitoring

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



Capgemini

What's next?

- Deep dive into Docker
- Advanced Swarm Mode
- Kubernetes
- Microservices in Practice (Deploy Spring Boot & Spring Cloud based microservices on Docker) - WKS 14.09.18, WRO 22-23.10.18
- ...

Capgemini

References

- Slides: https://github.com/ksobkowiak-talks/capgemini-craftsmanship-community-20180427/tree/master/slides
- Demo: https://github.com/ksobkowiak-talks/capgemini-craftsmanship-community-20180427/tree/master/demos

Capgemini

Credits

Special thanks to all the people who made and released their awesome resources for free:

- Docker for Java Developers by Fabiane Nardon and Arun Gupta
- Docker for Java Developers by Arun Gupta