

CRAFTSMANSHIP OPEN COMMUNITY SESSION #2

Capgemini Craftsmanship Community
Wrocław, 28th of April 2017

Krzysztof Sobkowiak (@ksobkowiak)

The Apache Software Foundation Member
Senior Solution Architect at Capgemini

YOUR SPEAKER

- FOSS enthusiast, evangelist & architect
- Senior Solution Architect at Capgemini
- The Apache Software Foundation
 - Member
 - Apache ServiceMix committer & PMC chair (V.P. Apache ServiceMix)
 - active at Apache Karaf, CXF, Camel, ActiveMQ
- Member/developer at OASP, OPS4J





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

APPSEVOLVE SUMMIT REPORT



- Mörfelden, 23-25 March 2017

- Service Orchestration Stream
 - Capgemini EiPaaS – current state
 - PaaS Shootout (EiPaaS, PCF, OpenShift, Fabric8)
- devonfw – a year after AppsEvolve 2016

COMING SOON

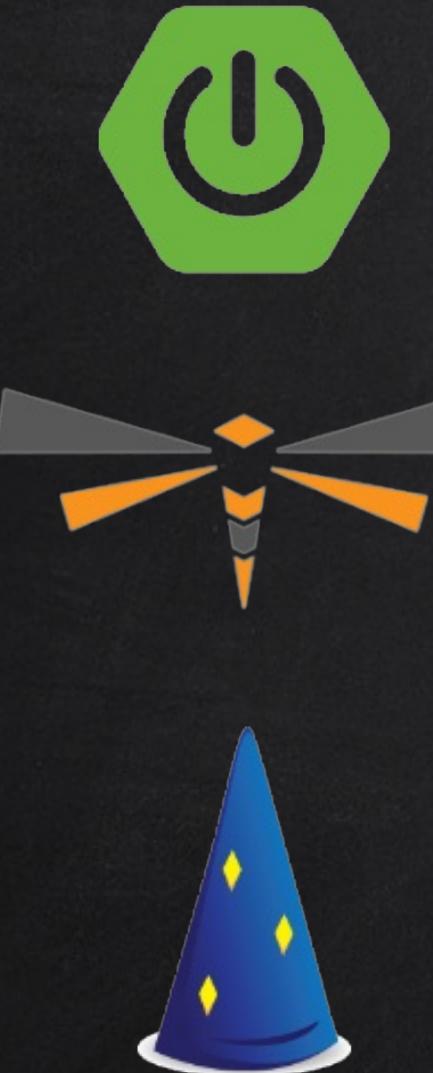
FABRIC8 YOUR SERVICES WITH KUBERNETES & OPENSHIFT

Craftsmanship

SOFT FLOWERS CRAFTSMANSHIP CLOTHING



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

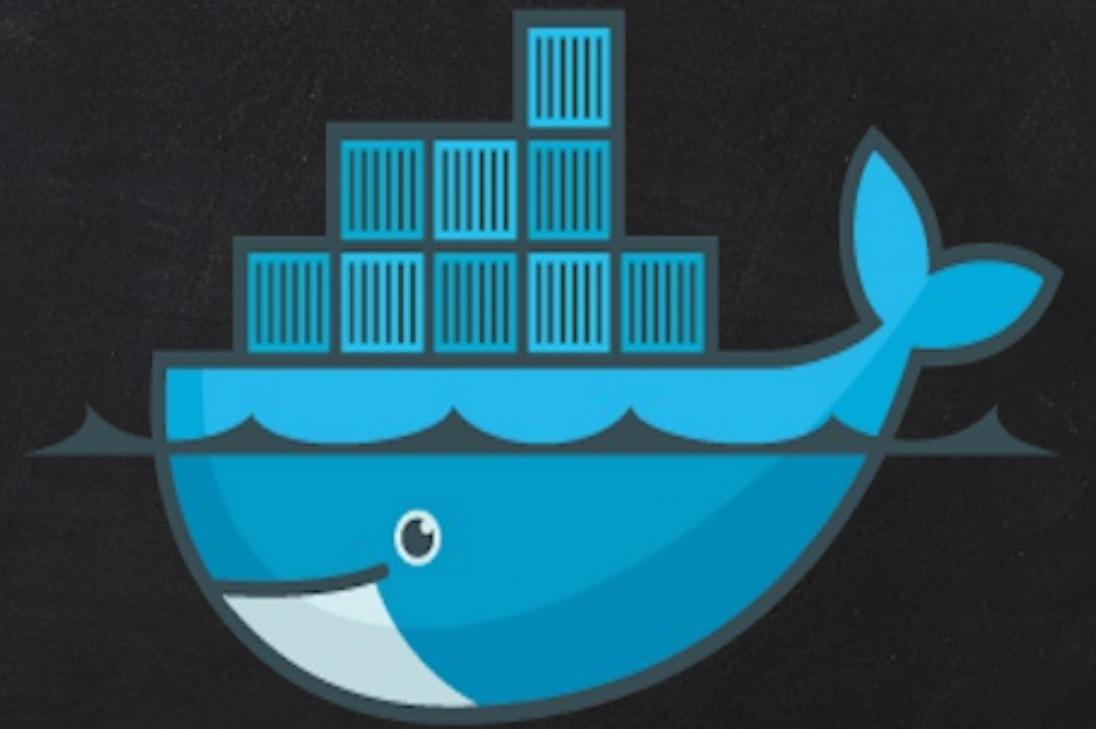


- Simple configuration
- Curated dependencies and transitive dependencies
- Built in metrics, monitoring
- Slim profile for deployment (...micro even?)

#microprofile

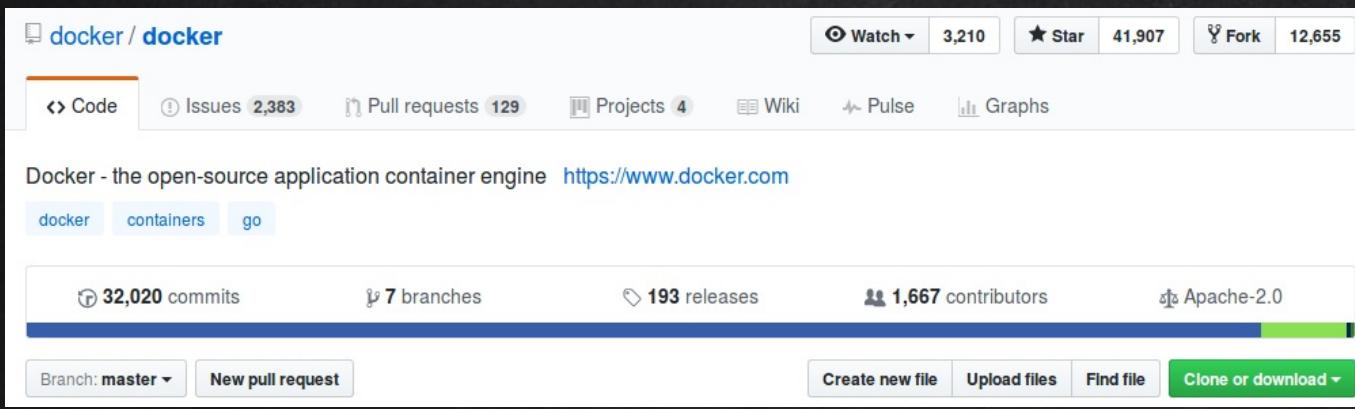


- Distributed configuration
- Service Discovery
- Loadbalancing
- Circuit Breakers
- Bulkheading
- Versioning/Routing
- Based on AWS

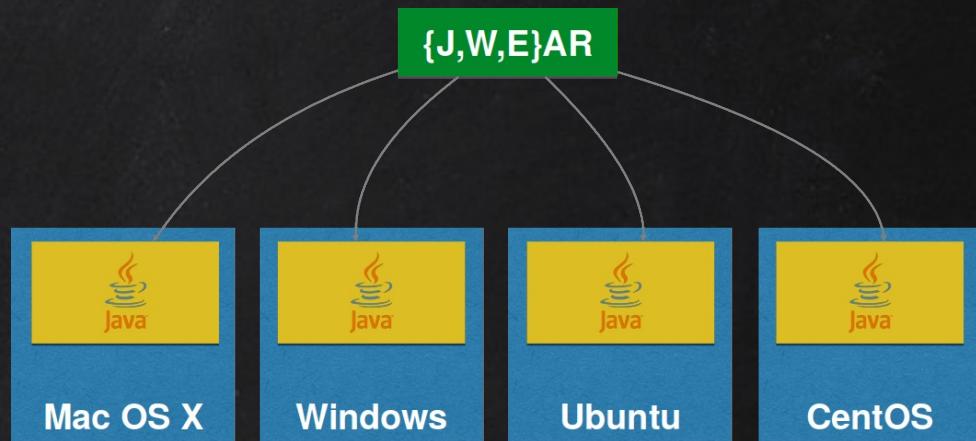


WHAT IS DOCKER?

- Open source project and company (not 100% true)



- WORA = Write Once Run Anywhere

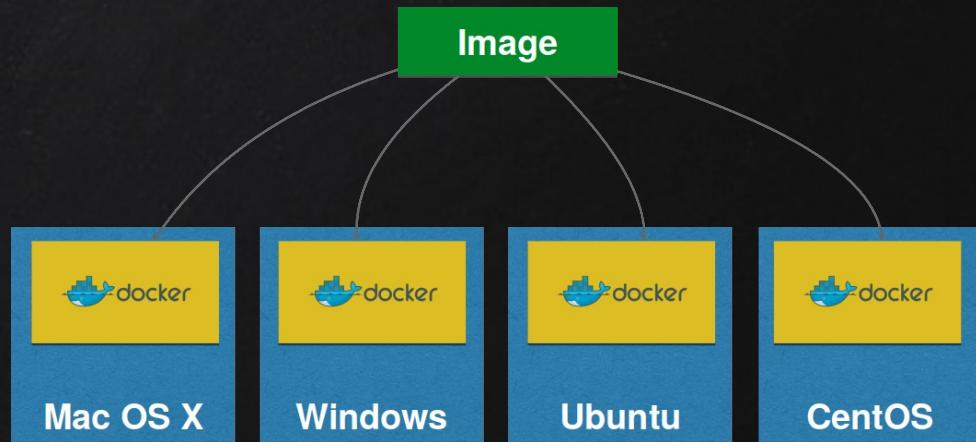


- Used to create containers for software applications

```
FROM openjdk  
  
COPY target/hello.jar /usr/src/hello.jar  
  
CMD java -cp /usr/src/hello.jar org.example.App
```

```
$ docker build -t username/hello  
$ docker run -it -p 8080:8080 username/hello
```

- PODA = Package Once Deploy Anywhere



ADDITIONAL DOCKER COMPONENTS

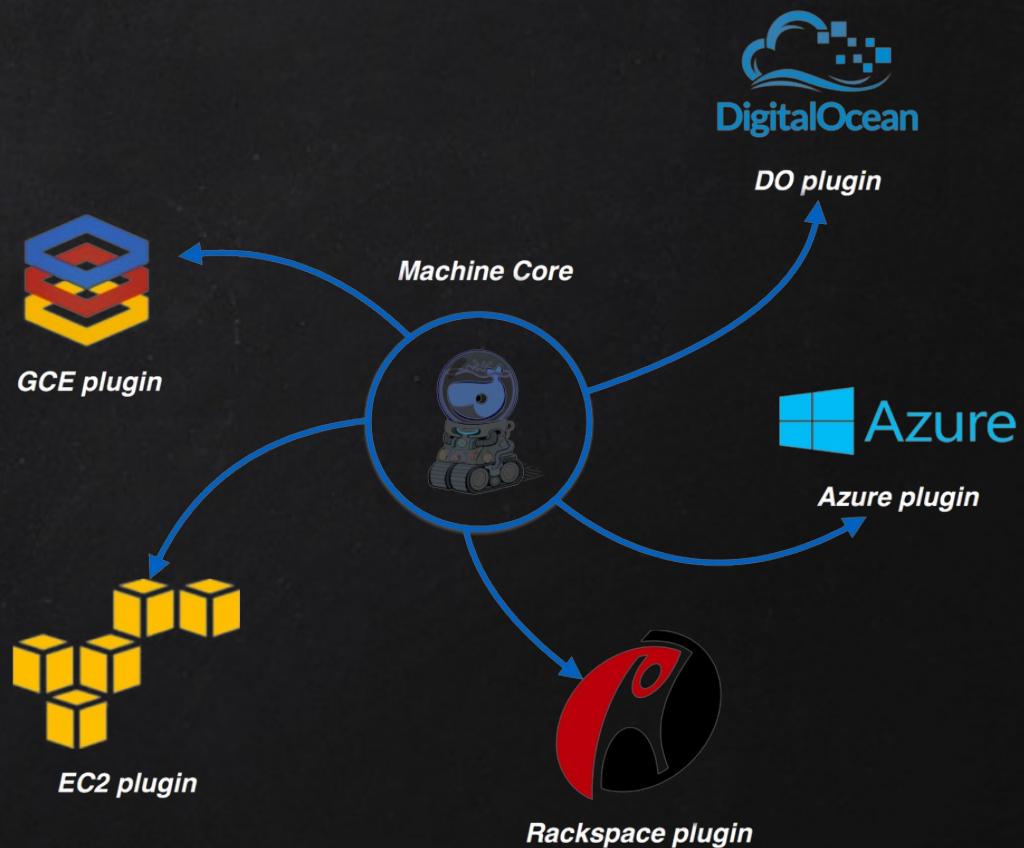
- Docker Compose

```
version: "3"
services:
  db:
    image: couchbase:latest
    ports:
      - 8091:8091
      - 8092:8092
      - 8093:8093
      - 11210:11210
  web:
    image: username/hello:latest
    environment:
      - COUCHBASE_URI=db
    ports:
      - 8080:8080
```

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

- Docker Machine

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



DEVOPS CHALLENGES FOR MULTIPLE CONTAINERS

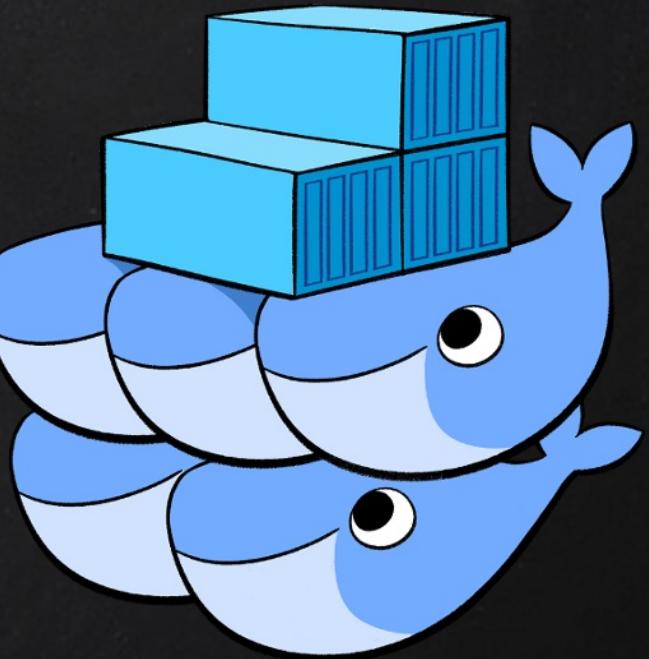
- How to scale?
- How to avoid port conflicts?
- How to manage them in multiple hosts?
- What happens if a host has a trouble?
- How to keep them running?
- How to update them?
- Where are my containers?

NETFLIX – WHAT ABOUT NON-JAVA?

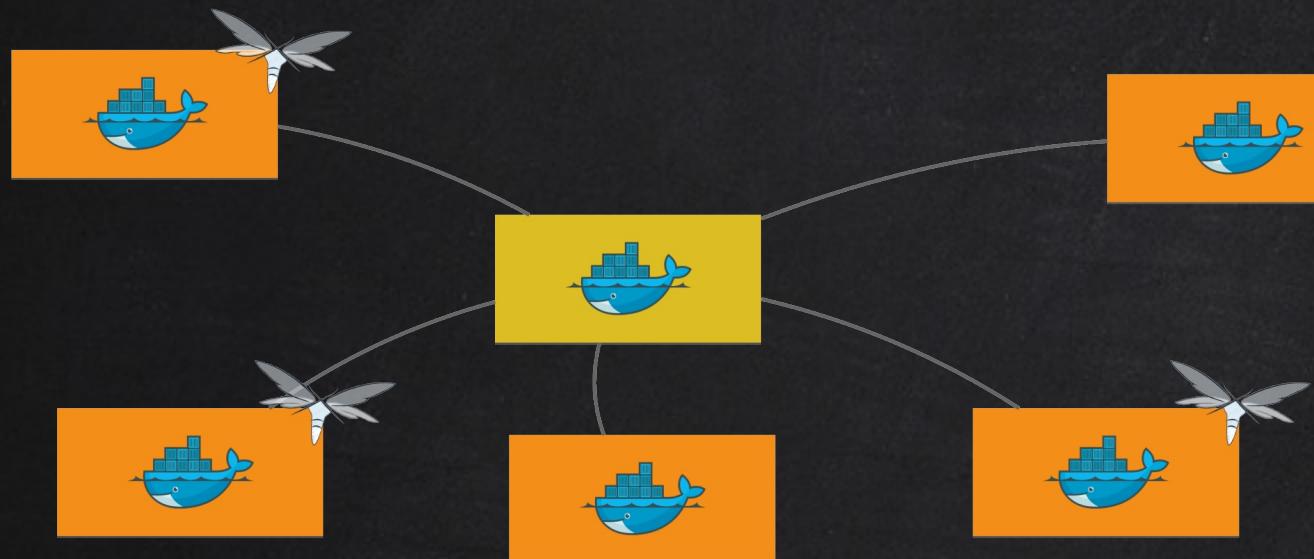


SWARM MODE

- 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



```
version: "3"
services:
  web:
    image: jboss/wildfly
    environment:
      - COUCHBASE_URI=db
    ports:
      - 8080:8080
    deploy:
      replicas: 3
      update_config:
        parallelism: 2
      restart_policy:
        condition: on-failure
```

```
$ docker service create --replicas 2 --name web jboss/wildfly
$ docker service scale web=3
```

```
$ docker stack deploy --compose-file=docker-compose.yml web
```

KUBERNETES



- Container orchestrator
- Supports multiple cloud and bare-metal environments
- Inspired by Google's experience with containers
- Provides:
 - Distributed configuration
 - Service Discovery, Loadbalancing
 - Versioning/Routing
 - Deployments, Scaling/Autoscaling
 - Liveness/Health checking, Self healing

KUBERNETES CONCEPTS

Pods

colocated group of containers that share an IP, namespace, storage volume, resources, lifecycle

Replica Set

manages the lifecycle of pods and ensures specified number are running (next gen Replication Controller)

Service

Single, stable name for a set of pods, also acts as LB

Label

used to organize and select group of objects

KUBERNETES CONCEPTS

Node

Machine or VM in the cluster

Master

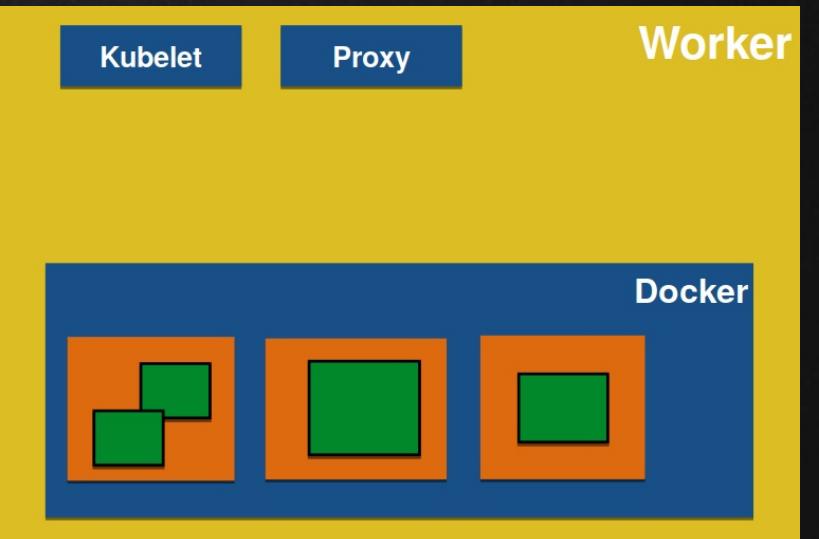
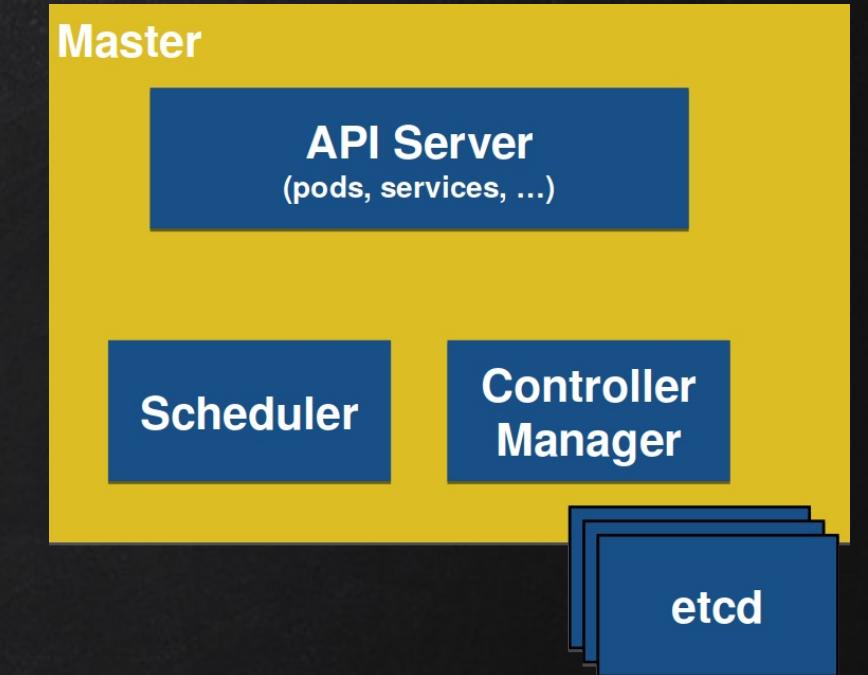
Central control plane, provides unified view of the cluster

- etcd: distributed key-value store used to persist Kubernetes system state

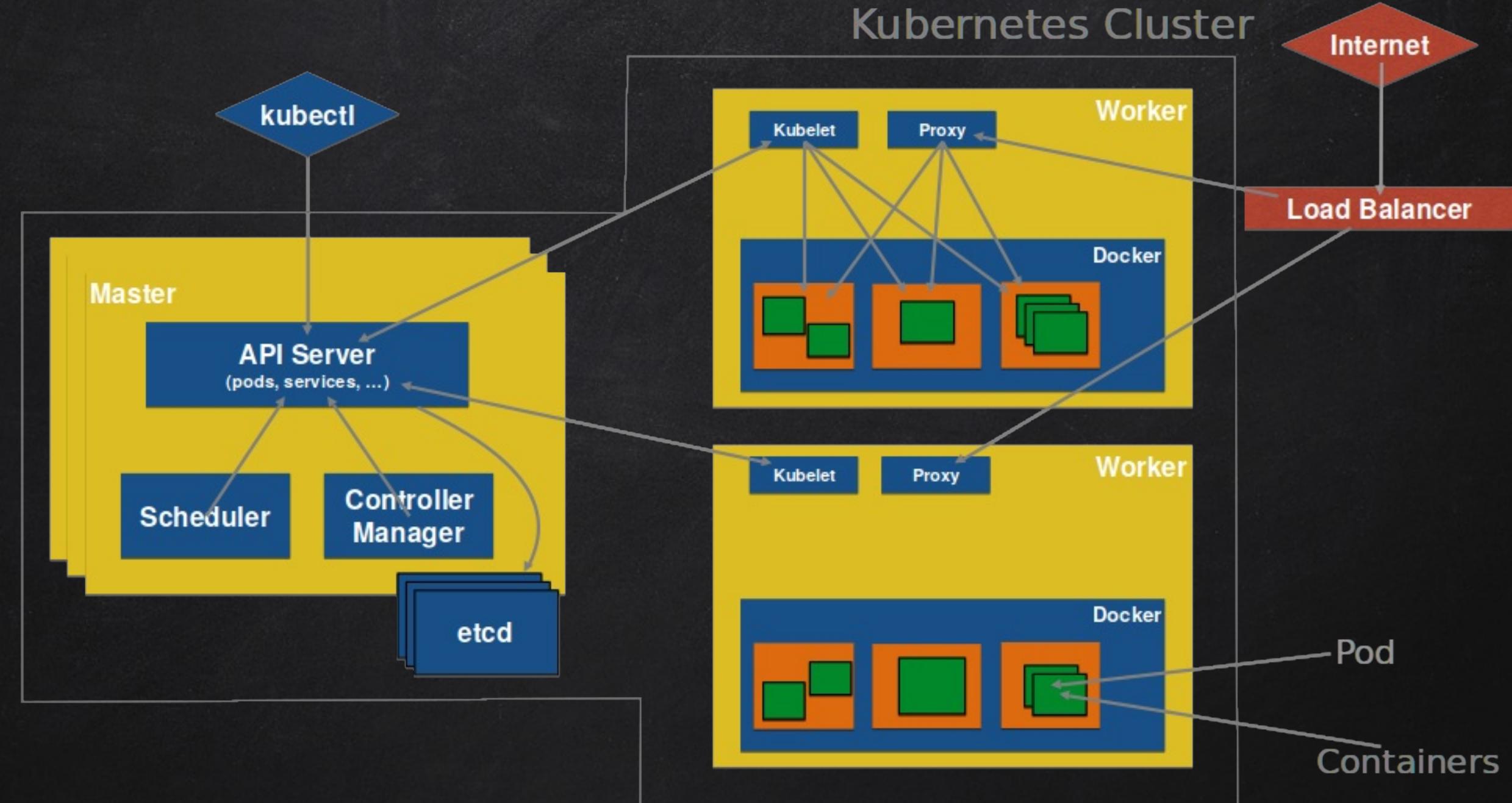
Worker

Docker host running kubelet (node agent) and proxy services

- Runs pods and containers



Kubernetes Cluster



OPENSHIFT IS KUBERNETES

- Team self service application deployment
- Developer workflow
- Enterprise focused (LDAP, RBAC, Oauth, etc)
- Integrated Docker registry
- Jenkins Pipeline out of the box
- Build/deployment triggers
- Software Defined Networking (SDN)
- Docker native format/packaging
- CLI/IDE/Web based tooling



FABRIC8 ALL THE THINGS!

- Built on top of Kubernetes
- Wizards to create microservices
- Package as immutable containers
- Rolling upgrade across environments
- 1-Click install of fully configured CI/CD (Jenkins Pipeline, Nexus, Git)
- Feedback loops
- Lots of developer tooling
- ChatOps
- iPaaS/Integration
- Chaos Monkey



TOPICS PROPOSAL

- Package your Application using Docker & Kubernetes
- Resilient Services with Spring Cloud, Docker & Kubernetes
- xPaaS with OpenShift & fabric8

TOPICS PROPOSAL

- Spring Boot
- WildFly Swarm
- Vert.x
- Dropwizard





THANKS!

Any questions?

You can find me at
@ksobkowiak
krzysztof.sobkowiak@capgemini.com
<http://krzysztof-sobkowiak.net>



This work is licensed under a Creative Commons Attribution 4.0 International License.