Docker, Kubernetes and Mesos for Java Developers

Arun Gupta, @arungupta
Docker Captain, Java Champion

©2016 Couchbase Inc.

What is Docker?

- Open source project and company

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

©2016 Couchbase Inc.

WORA = Write Once Run Anywhere

PODA = Package Once Deploy Anywhere

©2016 Couchbase Inc.

Docker Mission

©2016 Couchbase Inc.

©2016 Couchbase Inc.

Build

Develop an app using Docker containers with any language and any toolchain.

```
FROM ubuntu

CMD echo “Hello world”
```

```
FROM java

COPY target/hello.jar /usr/src/hello.jar

CMD java -cp /usr/src/hello.jar org.example.App
```

©2016 Couchbase Inc.

- Dockerfile reference
  - Usage
  - Format
    - Environment replacement
    - .dockerignore file
  - FROM
  - MAINTAINER
  - RUN
    - Known issues (RUN)
  - CMD
  - LABEL
  - EXPOSE
  - ENV
  - ADD
  - COPY
  - ENTRYPOINT
    - Exec form ENTRYPOINT example
    - Shell form ENTRYPOINT example
  - VOLUME
  - USER
  - WORKDIR
  - ARG
  - ONBUILD
  - STOPSIGNAL
  - Dockerfile examples

©2016 Couchbase Inc.

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

Docker Workflow

©2016 Couchbase Inc.

Union File System

©2016 Couchbase Inc.

Image Layers - Couchbase

```
~ > docker images couchbase
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL SIZE
couchbase           latest              45abdd57689a        3 weeks ago         372 MB
~ > docker history couchbase
IMAGE               CREATED             CREATED BY                                      SIZE
COMMENT
45abdd57689a        3 weeks ago         /bin/sh -c #(nop) VOLUME [/opt/couchbase/var]   0 B
dd8c5611343d        3 weeks ago         /bin/sh -c #(nop) EXPOSE 11207/tcp 11210/tcp    0 B
30852bbad62b        3 weeks ago         /bin/sh -c #(nop) CMD ["couchbase-server"]      0 B
5537747ea12f        3 weeks ago         /bin/sh -c #(nop) ENTRYPOINT &{["/entrypoint.   0 B
e8a83a5448df        3 weeks ago         /bin/sh -c #(nop) COPY file:cbb44c9c65b64a9dc   182 B
18165b90fefa        3 weeks ago         /bin/sh -c #(nop) COPY file:34e32c52f0895191f   389 B
5f37b8bdc5a6        3 weeks ago         /bin/sh -c wget -N $CB_RELEASE_URL/$CB_VERSIO   212.1 MB
1a8da511d01b        3 weeks ago         /bin/sh -c groupadd -g 1000 couchbase && user   328.7 kB
d9b2222c39b4        3 weeks ago         /bin/sh -c #(nop) ENV CB_VERSION=4.0.0 CB_REL   0 B
815f08b3c781        3 weeks ago         /bin/sh -c apt-get update &&     apt-get inst   23.57 MB
fc38f156c0ea        3 weeks ago         /bin/sh -c #(nop) MAINTAINER Couchbase Docker   0 B
2a7a952931ec        3 weeks ago         /bin/sh -c #(nop) CMD ["/bin/bash"]             0 B
10f1b5844a9c        3 weeks ago         /bin/sh -c sed -i 's/^#\s*\(deb.*universe\)$/   1.911 kB
23c388b926b6        3 weeks ago         /bin/sh -c echo '#!/bin/sh' > /usr/sbin/polic   156.2 kB
b45376f323f5        3 weeks ago         /bin/sh -c #(nop) ADD file:4a9e089e81d6581a54   135.9 MB
```

©2016 Couchbase Inc.

Image Layers - Java

```
~ > docker images java
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
java                openjdk-8-jdk-alpine   78afabc37d4f        6 days ago          145.5 MB
java                8                   f298aed75633        6 days ago          642.9 MB
java                latest              f298aed75633        6 days ago          642.9 MB
~ > docker history java
IMAGE               CREATED             CREATED BY                                      SIZE
COMMENT
f298aed75633        6 days ago          /bin/sh -c /var/lib/dpkg/info/ca-certificates   418.2 kB
<missing>           6 days ago          /bin/sh -c set -x  && apt-get update  && apt-   349.2 MB
<missing>           6 days ago          /bin/sh -c #(nop) ENV CA_CERTIFICATES_JAVA_VE   0 B
<missing>           6 days ago          /bin/sh -c #(nop) ENV JAVA_DEBIAN_VERSION=8u7   0 B
<missing>           6 days ago          /bin/sh -c #(nop) ENV JAVA_VERSION=8u72         0 B
<missing>           6 days ago          /bin/sh -c #(nop) ENV JAVA_HOME=/usr/lib/jvm/   0 B
<missing>           6 days ago          /bin/sh -c {   echo '#!/bin/sh';   echo 'set    87 B
<missing>           6 days ago          /bin/sh -c #(nop) ENV LANG=C.UTF-8              0 B
<missing>           6 days ago          /bin/sh -c echo 'deb http://httpredir.debian.   61 B
<missing>           6 days ago          /bin/sh -c apt-get update && apt-get install    1.289 MB
<missing>           2 weeks ago         /bin/sh -c apt-get update && apt-get install    122.6 MB
<missing>           2 weeks ago         /bin/sh -c apt-get update && apt-get install    44.32 MB
<missing>           2 weeks ago         /bin/sh -c #(nop) CMD ["/bin/bash"]             0 B
<missing>           2 weeks ago         /bin/sh -c #(nop) ADD file:b5391cb13172fb513d   125.1 MB
```

©2016 Couchbase Inc.

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

©2016 Couchbase Inc.

Docker Machine Providers

©2016 Couchbase Inc.

Docker for Mac/Windows

- Native user interface and auto-update capability
- No VirtualBox!
  - OSX: xhyve VM using Hypervisor.framework
  - Windows: Hyper-V VM
- Better networking and filesystem mounting/notification

©2016 Couchbase Inc.

Docker Toolbox

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

©2016 Couchbase Inc.

https://www.docker.com/toolbox

Docker Compose - One Service

```
version: “2”
services:
  db:
   image: couchbase
   volumes:
     - ~/couchbase:/opt/couchbase/var
   ports:
     - 8091:8091
     - 8092:8092
     - 8093:8093
     - 11210:11210
```

©2016 Couchbase Inc.

Docker Compose - Two Services

©2016 Couchbase Inc.

Docker Compose

- Defining and running multi-container applications
- Configuration defined in one or more files
  - `docker-compose.yml` (default)
  - `docker-compose.override.yml` (default)
  - Multiple files specified using `-f`
  - All paths relative to base configuration file
- Great for dev, staging, and CI

©2016 Couchbase Inc.

Docker Compose - Two Services

```
version: “2”
services:
  db:
    container_name: "db"
    image: couchbase
    ports:
      - 8091:8091
      - 8092:8092
      - 8093:8093
      - 11210:11210
  web:
    image: arungupta/wildfly
    environment:
      - COUCHBASE_URI=db
    ports:
      - 8080:8080
      - 9990:9990
```

©2016 Couchbase Inc.

Overriding Services in Docker Compose

```
mywildfly:
  image: jboss/wildfly
  ports:
    - 8080:8080
```

docker-compose.yml

```
mywildfly:
  ports:
    - 9080:8080
```

docker-compose.override.yml

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

©2016 Couchbase Inc.

Dev/Prod with Compose

```
mycouchbase:
  container_name: "db-dev"
  image: arungupta/couchbase
  ports:
    - . . .
mywildfly:
  image: arungupta/wildfly
  environment:
    - COUCHBASE_URI=db-dev:8093
  ports:
    - 8080:8080
```

docker-compose.yml

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

```
mywildfly:
 environment:
    - COUCHBASE_URI=db-prod:8093
  ports:
    - 8080:80
mycouchbase:
  container_name: "db-prod"
```

production.yml

```
docker-compose up
-f docker-compose.yml
-f production.yml
-d
```

©2016 Couchbase Inc.

Multi-host Networking

©2016 Couchbase Inc.

Default Networks

```
docker network ls
NETWORK ID          NAME                DRIVER
8cf651cafbef        bridge              bridge
14e63204639e        none                null
96901337c96f        host                host
```

Network Name	Purpose
`bridge`	Default network that containers connect to
`none`	Container-specific networking stack
`host`	Adds container on hosts networking stack

©2016 Couchbase Inc.

Multi-host Networking

- Create virtual networks and attach containers
- **`Bridge`** network span single host
- **`Overlay`** network spans multiple hosts
  - Uses **`libnetwork`** (built-in VXLAN-based overlay network driver) and Docker's **`libkv`**
- Works with Swarm and Compose
- Pluggable: Calico, Cisco, Weave, . . .

©2016 Couchbase Inc.

docker network

```
docker network --help

Usage: docker network [OPTIONS] COMMAND [OPTIONS]

Commands:
  disconnect                Disconnect container from a network
  inspect                   Display detailed network information
  ls                        List all networks
  rm                        Remove a network
  create                    Create a network
  connect                   Connect container to a network
```

©2016 Couchbase Inc.

Application Code

```
version: “2”
services:
  db:
    container_name: "db"
    image: couchbase
    ports:
      - 8091:8091
      - 8092:8092
      - 8093:8093
      - 11210:11210
  web:
    image: arungupta/wildfly-admin
    environment:
      - COUCHBASE_URI=db
    ports:
      - 8080:8080
      - 9990:9990
```

```
CouchbaseCluster.create(System.getenv(“COUCHBASE_URI”));
```

©2016 Couchbase Inc.

Docker Swarm

- Native clustering for Docker
- Provides a unified interface to a pool of Docker hosts
- Fully integrated with Machine and Compose
- Serves the standard Docker API
- 1.1 - Ready for production
  - Reschedule containers when a node fails (experimental)
  - Better node management

Stress tested on 1000 EC2 nodes, ~30k containers

©2016 Couchbase Inc.

©2016 Couchbase Inc.

Scheduling Backends

- Based on CPU (`-c`), RAM (`-m`), number of containers
- `docker machine create –strategy <value>`
  - `spread` (default): node with least number of running containers
  - `binpack`: node with most number of running containers
  - `random`: mostly for debugging
- API for pluggable backends (e.g. Mesos) coming

©2016 Couchbase Inc.

Machine + Swarm + Compose

©2016 Couchbase Inc.

Persistent Storage

- Data volumes - used to persist data independent of container's lifecycle
- Multiple plugins: Flocker, Ceph, . . .

```
docker volume --help

Usage: docker volume [OPTIONS] [COMMAND]

Manage Docker volumes

Commands:
  create                    Create a volume
  inspect                   Return low-level information on a volume
  ls                        List volumes
  rm                        Remove a volume
```

©2016 Couchbase Inc.

Persistent Storage

Create a volume

```
docker volume create –name=data data
```

Run a container with the volume

```
docker run -it -v data:/opt/couchbase/var couchbase
```

©2016 Couchbase Inc.

©2016 Couchbase Inc.

©2016 Couchbase Inc.

©2016 Couchbase Inc.

©2016 Couchbase Inc.

©2016 Couchbase Inc.

```
mycouchbase:
  container_name: "db"
  image: arungupta/couchbase
  ports:
    - 8091:8091
    - 8092:8092
    - 8093:8093
    - 11210:11210
mywildfly:
  image: arungupta/wildfly-admin
  environment:
    - COUCHBASE_URI=db
  ports:
    - 8080:8080
    - 9990:9990
```

http://blog.arungupta.me/docker-machine-swarm-compose-couchbase-wildfly/

©2016 Couchbase Inc.

Docker Machine + Compose + Swarm

©2016 Couchbase Inc.

Docker Machine + Compose + Swarm

©2016 Couchbase Inc.

Docker Mission

©2016 Couchbase Inc.

Getting Started with Kubernetes

Arun Gupta, @arungupta
VP Developer Advocacy, Couchbase

Couchbase

233

DZone REFCARDZ

BROUGHT TO YOU IN PARTNERSHIP WITH MESOSPHERE

Get More Refcardz! Visit DZone.com/Refcardz

CONTENTS

» What Is Kubernetes?
» Kubernetes Architecture
» Starting With Kubernetes
» Run Your First Container
» Scale Applications...and more!

Kubernetes

BY ARUN GUPTA

WHAT IS KUBERNETES?

Kubernetes (kubernetes.io) is an open-source orchestration system for managing containerized applications across multiple hosts, providing basic mechanisms for the deployment, maintenance, and scaling of applications.

Kubernetes, or "k8s" or "kube" for short, allows the user to declaratively specify the desired state of a cluster using high-level primitives. For example, the user may specify that they want three instances of the Couchbase server container running. Kubernetes' self-healing mechanisms, such as auto-restarting, re-scheduling, and replicating containers then converge the actual state towards the desired state.

Kubernetes supports Docker and Rocket containers. An abstraction around the containerization layer will allow for other container image formats and runtimes to be supported in the future.

KEY CONCEPTS OF KUBERNETES

POD

A Pod is the smallest deployable unit that can be created, scheduled, and managed. It's a logical collection of containers that belong to an application.

Each resource in Kubernetes is defined using a configuration file. For example, a Couchbase pod can be defined with the following `.yaml` file:

A Replication Controller creating two instances of a Couchbase pod can be defined as:

```
apiVersion: v1
kind: ReplicationController
metadata:
  name: couchbase-controller
spec:
  # Two replicas of the Pod to be created
  replicas: 2
  # Identifies the label key and value on the Pod that
  # this Replication Controller is responsible for
  managing
  selector:
    app: couchbase-rc-pod
  # 'cookie cutter' used for creating new pods when
  necessary
  template:
    metadata:
      labels:
        # label key and value on the pod.
        # These must match the selector above.
        app: couchbase-rc-pod
    spec:
      containers:
      - name: couchbase
        image: couchbase
        ports:
        - containerPort: 8091
```

SERVICE

Each Pod is assigned a unique IP address. If the Pod is inside a Replication Controller, then the pod is recreated but may be given

dzone.com/refcardz/kubernetes-essentials

©2016 Couchbase Inc.

Kubernetes

- Open source orchestration system for Docker containers
- Provide declarative primitives for the “desired state”
  - Self-healing
  - Auto-restarting
  - Schedule across hosts
  - Replicating

©2016 Couchbase Inc.

Concepts

- **Pods**: collocated group of Docker containers that share an IP and storage volume
- **Service**: Single, stable name for a set of pods, also acts as LB
- **Label**: used to organize and select group of objects
- **Replication Controller**: manages the lifecycle of pods and ensures specified number are running

©2016 Couchbase Inc.

Components

- **Node**: Docker host running *kubelet* (node agent) and *proxy* services
  - Monitored by *systemd* (CentOS) or *monit* (Debian)
- **Master**: hosts cluster-level control services, including the API server, scheduler, and controller manager
- **etcd**: distributed key-value store used to persist Kubernetes system state

©2016 Couchbase Inc.

Architecture

©2016 Couchbase Inc.

©2016 Couchbase Inc.

Master High Availability

- Hack by running a `podmaster` utility
- Proposal
  - Hot Standby
  - Warm Standby
  - Active-Active (Load Balanced)

©2016 Couchbase Inc.

http://kubernetes.io/v1.1/docs/proposals/high-availability.html

kubectl

- Controls the Kubernetes cluster manager
- `kubectl get pods` or `minions`
- `kubectl create -f <filename>`
- `kubectl update` or `delete`
- `kubectl resize –replicas=3 replicationcontrollers <name>`

©2016 Couchbase Inc.

Kubernetes Config

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: wildfly-pod
  labels:
    name: wildfly
spec:
  containers:
    - image: jboss/wildfly
      name: wildfly-pod
      ports:
        - containerPort: 8080
```

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: wildfly-rc
  labels:
    name: wildfly
spec:
  replicas: 2
  template:
    metadata:
      labels:
        name: wildfly
    spec:
      containers:
      - name: wildfly-rc-pod
        image: jboss/wildfly
        ports:
        - containerPort: 8080
```

©2016 Couchbase Inc.

```
export KUBERNETES_PROVIDER=vagrant
./cluster/kube-up.sh
```

©2016 Couchbase Inc.

©2016 Couchbase Inc.

https://get.k8s.io/

A Pod with One Container

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: wildfly-pod
  labels:
    name: wildfly
spec:
  containers:
    - image: jboss/wildfly
      name: wildfly-pod
      ports:
        - containerPort: 8080
```

©2016 Couchbase Inc.

Services

- Abstract a set of pods as a single IP and port
  - Simple TCP/UDP load balancing
- Creates environment variables in other pods
- Stable endpoint for pods to reference
  - Allows list of pods to change dynamically

©2016 Couchbase Inc.

Services

©2016 Couchbase Inc.

Services

©2016 Couchbase Inc.

Services - AWS

©2016 Couchbase Inc.

Services - GCE

©2016 Couchbase Inc.

Services - Azure

©2016 Couchbase Inc.

Services

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: couchbase-rc
  labels:
    name: couchbase-rc
    context: docker-k8s-lab
spec:
  replicas: 1
  template:
    metadata:
      name: couchbase-rc-pod
      labels:
        name: couchbase-rc-pod
        context: docker-k8s-lab
    spec:
      containers:
      - name: couchbase-rc-pod
        image: couchbase/server
        volumeMounts:
        - mountPath: "/opt/couchbase/var"
          name: mypd
        ports:
        - containerPort: 8091
          hostPort: 8091
        - containerPort: 8092
        - containerPort: 8093
        - containerPort: 11210
        -
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: couchbase-service
  labels:
    name: couchbase-service-pod
    context: docker-k8s-lab
spec:
  ports:
    - port: 8091
      port: 8092
      port: 8093
      port: 11210
  # label keys and values that must
  selector:
    name: couchbase-rc-pod
    context: docker-k8s-lab
```

©2016 Couchbase Inc.

Service across Two Nodes

©2016 Couchbase Inc.

Replication Controller

- Ensures that a specified number of pod "replicas" are running
  - Pod templates are cookie cutters
  - Rescheduling
  - Manual or auto-scale replicas
  - Rolling updates
- Recommended to wrap a Pod or Service in a RC
- Only appropriate for Pods with `Restart=Always` policy (default)

©2016 Couchbase Inc.

Replication Controller Configuration

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: wildfly-rc
  labels:
    name: wildfly
    context: docker-k8s-lab
spec:
  replicas: 1
  template:
    metadata:
      labels:
        name: wildfly
    spec:
      containers:
      - name: wildfly-rc-pod
        image: arungupta/wildfly-mysql-javaee7:k8s
        ports:
        - containerPort: 8080
```

©2016 Couchbase Inc.

Replication Controller

©2016 Couchbase Inc.

Replication Controller: Automatic Rescheduling

©2016 Couchbase Inc.

Replication Controller: Scaling

©2016 Couchbase Inc.

Sample Production Deployment

- www.wombatsoftware.de
- shopadvisors.de: E-commerce optimization and monitoring tools for increase of sales

©2016 Couchbase Inc.

Sample Production Deployment

Load	Containers
Normal	400
Peak	600

©2016 Couchbase Inc.

Health Checks

- Restarts Pod, if wrapped in RC
- Application-level health checks
  - HTTP
  - Container Exec
  - TCP Socket
- Health checks performed by Kubelet

©2016 Couchbase Inc.

Kubernetes using Docker

```
etcd:
  image: gcr.io/google_containers/etcd:2.0.9
  net: "host"
  entrypoint: /usr/local/bin/etcd --addr=127.0.0.1:4001 --bind-addr=0.0.0.0:4001 --data-dir=/v
master:
  image: gcr.io/google_containers/hyperkube:v0.21.2
  net: "host"
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  entrypoint: /hyperkube kubelet --api_servers=http://localhost:8080 --v=2 --address=0.0.0.0 -
proxy:
  image: gcr.io/google_containers/hyperkube:v0.21.2
  net: "host"
  privileged: true
  entrypoint: /hyperkube proxy --master=http://127.0.0.1:8080 --v=2
```

©2016 Couchbase Inc.

https://github.com/arun-gupta/docker-images/blob/master/kubernetes/docker-compose.yml

Kubernetes using Docker

©2016 Couchbase Inc.

http://kubernetes.io/v1.0/docs/getting-started-guides/docker.html

OpenShift 3

©2016 Couchbase Inc.

References

- github.com/javaee-samples/docker-java
- kubernetes.io
- Containers recipe: couchbase.com/containers

©2016 Couchbase Inc.

Getting Started with Mesos

Arun Gupta, @arungupta
VP Developer Advocacy, Couchbase

Couchbase

Mesos

- Open source cluster manager
- Developed at UC Berkeley
- Provides resource isolation and sharing across distributed applications
- Run distributed systems on the same pool of nodes
  - Hadoop, Spark, Jenkins, Couchbase, …
- Cluster monitoring
- Tasks isolated via Linux containers

©2016 Couchbase Inc.

Mesos

- Master Slave architecture
- Fault tolerant
  - Leader election via ZooKeeper
- Multi platform
  - Ubuntu, Mac OS, CentOS

©2016 Couchbase Inc.

Mesos Architecture

©2016 Couchbase Inc.

http://mesos.apache.org/documentation/latest/architecture/

©2016 Couchbase Inc.

http://mesos.apache.org/documentation/latest/architecture/

Frameworks

- Frameworks are targeted at a use case and domain-specific
  - Master node “offers” resources to each framework
  - Framework “accepts” the offer and execute applications
- Framework has “scheduler” and “executor”
  - Scheduler registers with the master for “offer”
  - Executor launched on slave nodes to run the task
    - Passes a description of the task to run

©2016 Couchbase Inc.

©2016 Couchbase Inc.

http://mesos.apache.org/documentation/latest/architecture/

docker

kubernetes

MESOS

©2016 Couchbase Inc.

References

- github.com/javaee-samples/docker-java
- mesos.apache.org/
- Containers recipe: couchbase.com/containers

©2016 Couchbase Inc.