Big Data Engineer Bootcamp

Code 1

Agenda

Project Structure

Dev Environment

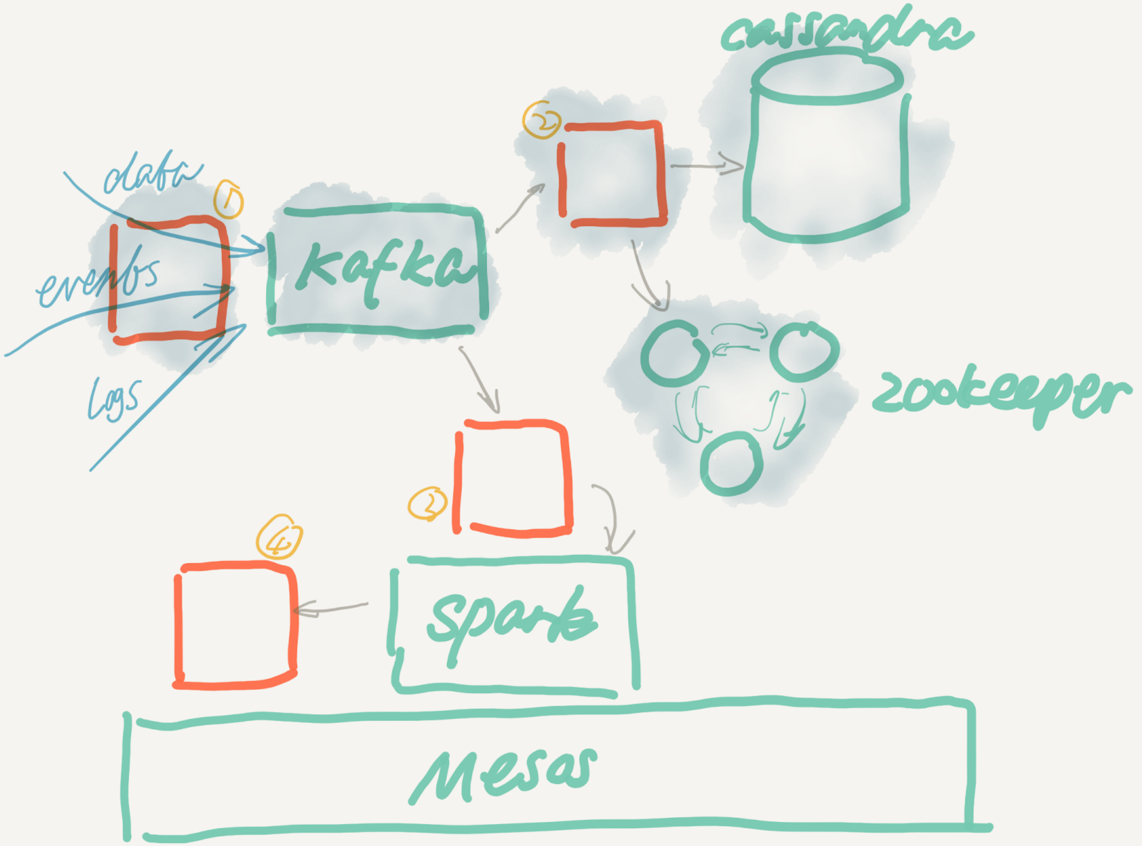
Work with Zookeeper

Work with Kafka

Work with Cassandra

Project Structure

- Apache Kafka
- Apache Zookeeper
- Apache Cassandra



Agenda

Project Structure

Dev Environment

Work with Zookeeper

Work with Kafka

Work with Cassandra

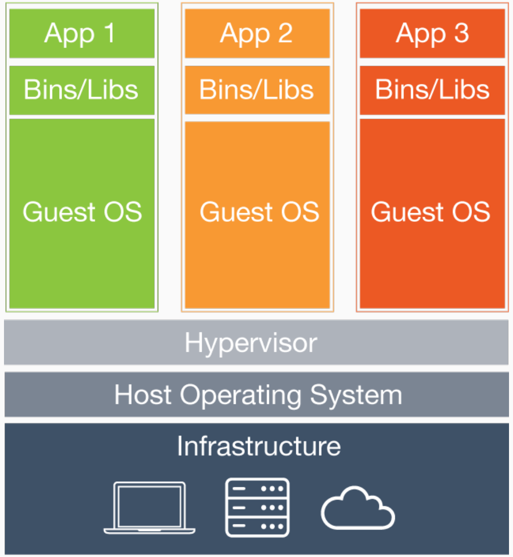
Requirement

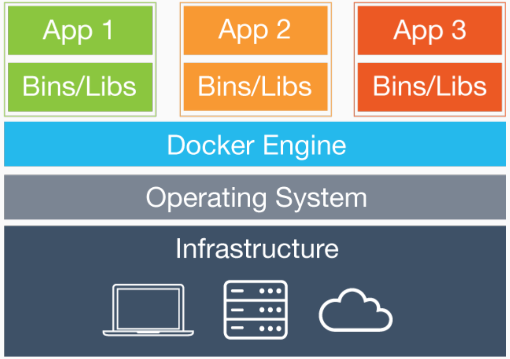
- Docker
- Virtualbox
- Docker-machine
  - 请一定要安装，不然会导致环境不一致
- wget
- tar
- Scala
- SBT
- Python
- Pip

What is Docker

- A tool to package and deploy applications inside containers
  - Containers are isolated environments
- Developed by Solomon Hykes in Dotcloud
- Open-sourced in March 2013
- Grown into a platform
  - Docker Compose, Docker Swarm
  - Docker Image Hosting
  - Container Hosting

Why Docker


App 1
App 2
App 3
Bins/Libs
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Guest OS
Hypervisor
Host Operating System
Infrastructure


App 1
App 2
App 3
Bins/Libs
Bins/Libs
Bins/Libs
Docker Engine
Operating System
Infrastructure

Docker Internal

- Client - Server Architecture
  - Docker command line = client 客户端可以连接任何远端的docker daemon
  - Docker daemon = server
- Use Linux functionalities (unavailable on MacOS or Windows)
  - cgroup
  - namespaces
  - Etc

Docker on Mac/Docker Machine

- 请一定要安装docker machine

- Docker on Mac默认会在本地启动一个小型的虚拟机，会导致后面的命令代码不一致

Play with Docker

- `docker-machine create --driver virtualbox --virtualbox-cpu-count 2 --virtualbox-memory 2048 bigdata`
    - 创建一个名字叫做bigdata的虚拟机
    - 在这次课中，请一定要用docker machine创建虚拟机再运行docker命令，不然会导致命令无法正常执行
- `Docker-machine ip bigdata`
    - 这个命令可以帮助你看到bigdata这个虚拟机的ip地址
    - 后面的很多命令咱们都会用这个命令来获取正确的ip地址
- `eval $(docker-machine env bigdata)`
    - 注意这一个命令是必需的，能够帮助你的docker客户端跟服务器通信，每一个新的terminal窗口都需要输入这个命令
- `docker run -d -p 3000:3000 unclebarney/chit-chat`
- `docker images`
- `docker ps`

Docker on Windows/Docker Machine

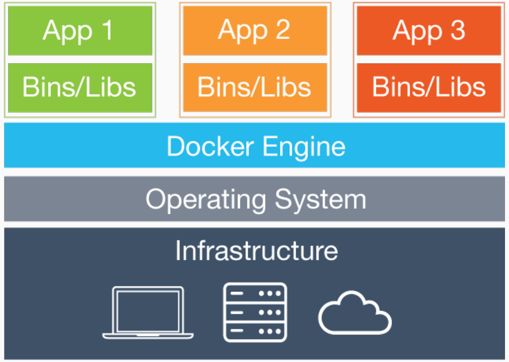
- 安装Docker Toolbox (Include: VirtualBox, Docker-machiine, Docker-compose)
  - Instruction: https://docs.docker.com/toolbox/toolbox_install_windows/
  - Download: https://www.docker.com/products/docker-toolbox
- 安装好后，新开一个terminal，然后使用docker-machine ls确认能否正确运行
  - 会列出已有的虚拟机

Play with Docker-Windows

- docker-machine create --driver virtualbox --virtualbox-cpu-count 2 --virtualbox-memory 2048 bigdata
    - 创建一个名字叫做bigdata的虚拟机
    - 在这次课中，请一定要用docker machine创建虚拟机再运行docker命令，不然会导致命令无法正常执行
- Docker-machine ip bigdata
    - 这个命令可以帮助你看到bigdata这个虚拟机的ip地址
    - 后面的很多命令咱们都会用这个命令来获取正确的ip地址
- docker-machine env --shell cmd default
- FOR /f "tokens=*" %i IN ('docker-machine env --shell cmd bigdata') DO %i
    - 注意这个命令是必需的，能够帮助你的docker客户端跟服务器通信，每一个新的terminal窗口都需要输入这个命令
- docker run -d -p 3000:3000 unclebarney/chit-chat
- docker images
- docker ps

Dev Environment

- Run all the servers/components as docker container
- For example
  - App 1 = zookeeper
  - App 2 = kafka
  - App 3 = cassandra



- Allow fast iteration

Agenda

Project Structure

Dev Environment

Work with Zookeeper

Work with Kafka

Work with Cassandra

Start Zookeeper Server

- `docker run -d -p 2181:2181 -p 2888:2888 -p 3888:3888 --name zookeeper confluent/zookeeper`
- `docker images`
- `docker ps`

Get Zookeeper CLI

- Download using shell commands (MacOS, Linux, Unix)
  - wget http://apache.mirrors.ionfish.org/zookeeper/zookeeper-3.4.8/zookeeper-3.4.8.tar.gz
  - tar xvf zookeeper-3.4.8.tar.gz
  - mv zookeeper-3.4.8 zookeeper
  - rm zookeeper-3.4.8.tar.gz
- Download directly (Windows)
  - http://www.apache.org/dyn/closer.cgi/zookeeper/

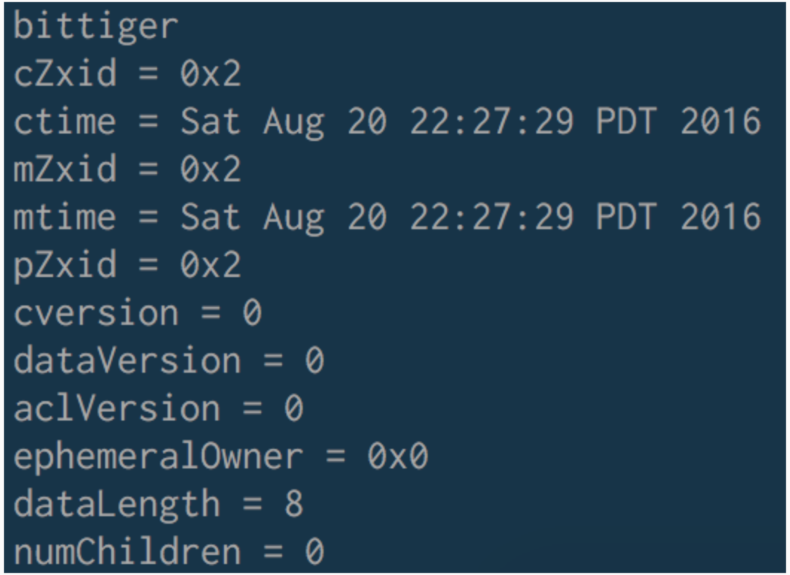
- cd zookeeper/bin (MacOS, Linux, Unix)
- ./zkCli.sh -server `docker-machine ip bigdata`:2181
- ./zkCli.sh -server localhost:2181
- cd zookeeper/bin (Windows)
- docker-machine ip bigdata
  - memorize virtual machine ip (ex. 192.168.99.100)
- zkCli.cmd -server 192.168.99.100:2181

Browse Znode Data

- `ls /`
- `ls /zookeeper`
- `get /zookeeper/quota`

Create Znode Data

- `create /workers "bittiger"`
- `ls /`
- `ls /workers`
- `get /workers`



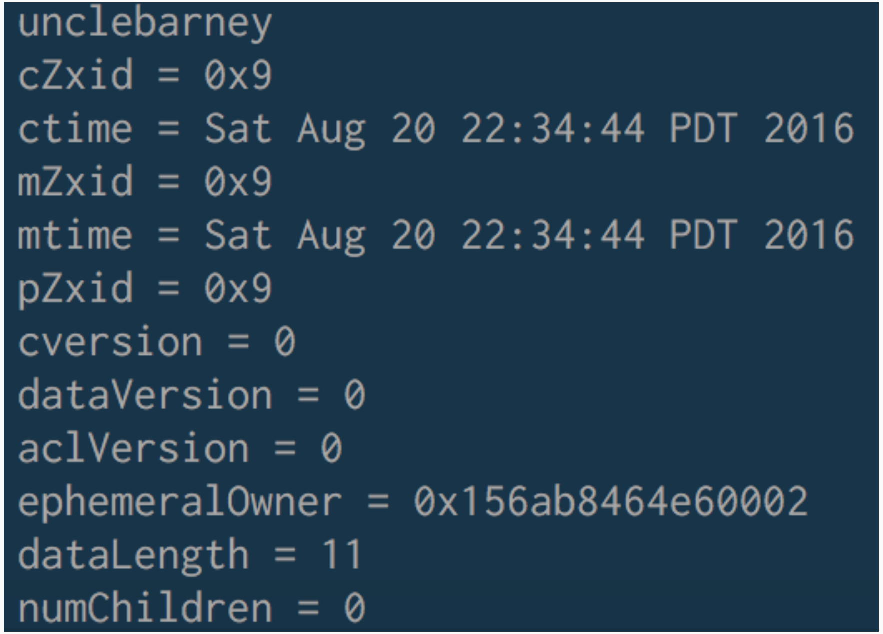
```
bittiger
cZxid = 0x2
ctime = Sat Aug 20 22:27:29 PDT 2016
mZxid = 0x2
mtime = Sat Aug 20 22:27:29 PDT 2016
pZxid = 0x2
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x0
dataLength = 8
numChildren = 0
```

Delete Znode Data

- `delete /workers`
- `ls /`
- `ls /workers`
- `get /workers`

Create Ephemeral Znode Data

- `create -e /workers "unclebarney"`
- `ls /`
- `ls /workers`
- `get /workers`



```
unclebarney
cZxid = 0x9
ctime = Sat Aug 20 22:34:44 PDT 2016
mZxid = 0x9
mtime = Sat Aug 20 22:34:44 PDT 2016
pZxid = 0x9
cversion = 0
dataVersion = 0
aclVersion = 0
ephemeralOwner = 0x156ab8464e60002
dataLength = 11
numChildren = 0
```

Watcher

- `get /workers true`

Agenda

Project Structure

Dev Environment

Work with Zookeeper

Work with Kafka

Work with Cassandra

Dependencies

- `scala -version`
- `sbt --version`
- `python --version`
- `pip --version`

Start Kafka Server

-
  ```
  docker run -d -p 9092:9092 -e KAFKA_ADVERTISED_HOST_NAME=`docker-machine ip bigdata` -e KAFKA_ADVERTISED_PORT=9092 --name kafka --link zookeeper:zookeeper confluent/kafka
  ```

-
  ```
  docker images
  ```

-
  ```
  docker ps
  ```

Get Kafka CLI

- Download using command line (MacOS, Unix, Linux)
  - wget http://apache.mirrors.ionfish.org/kafka/0.10.0.1/kafka_2.11-0.10.0.1.tgz
  - tar xvf kafka_2.11-0.10.0.1.tgz
  - mv kafka_2.11-0.10.0.1 kafka
  - rm kafka_2.11-0.10.0.1.tgz
- Download directly (Windows)
  - https://www.apache.org/dyn/closer.cgi?path=/kafka/0.10.0.1/kafka_2.11-0.10.0.1.tgz

Create Kafka Topic

- (MacOS, Unix, Linux)
- ./kafka-topics.sh --create --zookeeper `docker-machine ip bigdata` --replication-factor 1 --partitions 1 --topic bigdata
- ./kafka-topics.sh --list --zookeeper `docker-machine ip bigdata`

- (Windows)
- docker-machine ip bigdata
    - memorize virtual machine ip (ex. 192.168.99.100, please change accordingly. I will use this in all following slices for **windows**)
- ./windows/kafka-topics.bat --create --zookeeper `docker-machine ip bigdata` --replication-factor 1 --partitions 1 --topic bigdata
    - problem of “java.lang.classnotfoundException” please look up classpath in environ var
- ./windows/kafka-topics.bat --list --zookeeper `docker-machine ip bigdata`

Look up on Zookeeper

- (MacOS, Unix, Linux)
- ./zkCli.sh -server `docker-machine ip bigdata`:2181
- ls /

- (Windows)
- zkCli.cmd -server 192.168.99.100:2181
- ls /

Produce Messages

- (MacOS, Unix, Linux)
- `./kafka-console-producer.sh --broker-list `docker-machine ip bigdata`:9092 --topic bigdata`

- (Windows)
- `kafka-console-producer.bat --broker-list 192.168.99.100:9092 --topic bigdata`

Consume Messages

- (MacOS, Unix, Linux)
-
  ```
  ./kafka-console-consumer.sh --zookeeper `docker-machine ip bigdata`:2181 --topic bigdata
  ```
-
  ```
  ./kafka-console-consumer.sh --zookeeper `docker-machine ip bigdata`:2181 --topic bigdata --from-beginning
  ```

- (Windows)
-
  ```
  kafka-console-consumer.bat --zookeeper 192.168.99.100:9092:2181 --topic bigdata
  ```
-
  ```
  kafka-console-consumer.bat --zookeeper 192.168.99.100:9092:2181 --topic bigdata --from-beginning
  ```

Look Into Kafka Broker

- `docker exec -it kafka bash`
- `cd /var/lib/kafka`
- `ls`

Agenda

Project Structure

Dev Environment

Work with Zookeeper

Work with Kafka

Work with Cassandra

Start Cassandra Server

-
  ```
  docker run -d -p 7199:7199 -p 9042:9042 -p 9160:9160 -p 7001:7001 --name cassandra cassandra:3.7
  ```

-
  ```
  docker images
  ```

-
  ```
  docker ps
  ```

Get Cassandra CLI

- Download using command line (Mac, Linux, Unix)
    - wget http://apache.mirrors.ionfish.org/cassandra/3.7/apache-cassandra-3.7-bin.tar.gz
    - tar xvf apache-cassandra-3.7-bin.tar.gz
    - mv apache-cassandra-3.7 cassandra
    - rm apache-cassandra-3.7-bin.tar.gz
- Direct Download(windows)
    - http://www.apache.org/dyn/closer.lua/cassandra/3.7/apache-cassandra-3.7-bin.tar.gz

- ./cqlsh `docker-machine ip bigdata` 9042(Mac, Linux, Unix)
- cqlsh.bat 192.168.99.100 9042(windows)

Create Keyspace’

-
  ```
  ./cqlsh `docker-machine ip bigdata` 9042
  ```

-
  ```
  CREATE KEYSPACE “stock” WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1} AND durable_writes = 'true';
  ```

-
  ```
  USE stock;
  ```

-
  ```
  DESCRIBE KEYSPACE;
  ```

Create Table

-
  ```
  ./cqlsh `docker-machine ip bigdata` 9042
  ```

-
  ```
  CREATE TABLE user ( first_name text, last_name text, PRIMARY KEY (first_name));
  ```

-
  ```
  DESCRIBE TABLE user;
  ```

Insert Data

```
./cqlsh `docker-machine ip bigdata` 9042
```

```
INSERT INTO user (first_name, last_name) VALUES ('uncle', 'barney');
```

Query Data

- ./cqlsh `docker-machine ip bigdata` 9042
- SELECT COUNT (*) FROM USER;
- SELECT * FROM user WHERE first_name='uncle';
- SELECT * FROM user WHERE last_name='barney';

Look Into Cassandra Node

- `docker exec -it cassandra bash`
- `cd /var/lib/cassandra`
- `ls`

Delete Data

-
  ```
  ./cqlsh `docker-machine ip bigdata` 9042
  ```

-
  ```
  DELETE last_name FROM user WHERE first_name='uncle';
  ```

-
  ```
  DELETE FROM user WHERE first_name='uncle';
  ```

Remove Table

```
./cqlsh `docker-machine ip bigdata` 9042
```

```
TRUNCATE user;
```

```
DROP TABLE user;
```