Distributed System in Docker using DevOps

VESIT | 1st March 2018 | Mohan Pawar

Agenda

- Distributed Systems - Brief
- Docker Overview
- Docker Fundamentals
- Deployment using Docker
- Future Scope

What is DevOps ?

Docker is the world's leading software containerization platform.

What is Distributed System ?

"A collection of independent computers that appear to its users as one computer."

- ***Andrew Tannenbaum***

Make your workstation Docker Ready !

Install Docker for Mac, Windows and Linux. **Community Edition(CE)**

Enterprise Edition(EE)

Docker

Cloud Install

aws Azure DigitalOcean

The Best way is installing docker from source.

Three Characteristics

- The computers operate concurrently.
- The computers fail independently.
- The computers don’t share a global clock.

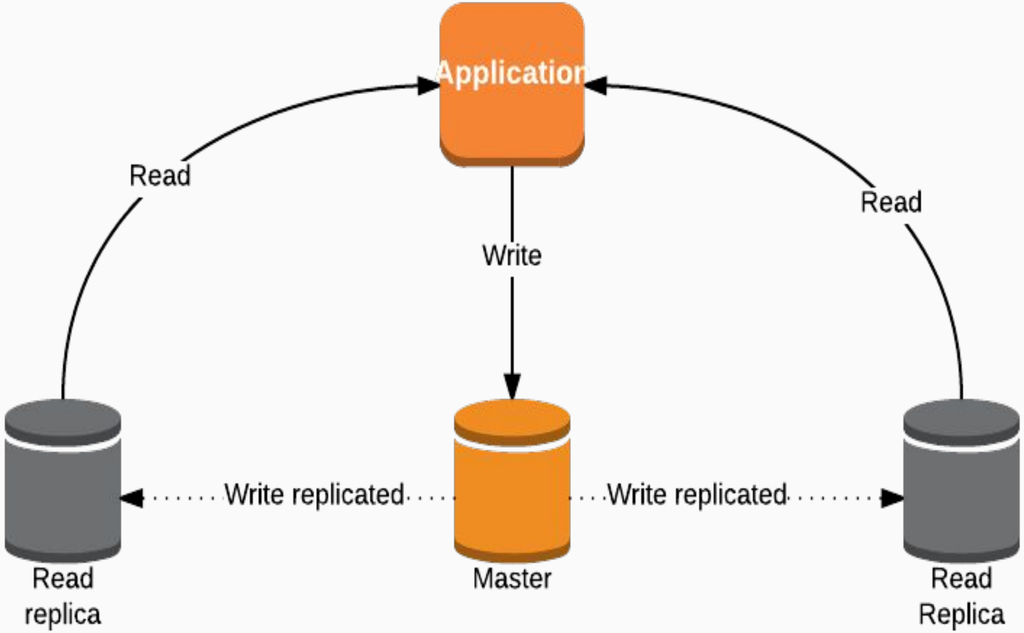
Three Topics

- Storage
- Computation
- Messaging

Distributed Storage

SIngle-Master Storage

Read Replication



Inconsistency Problem

Sharding

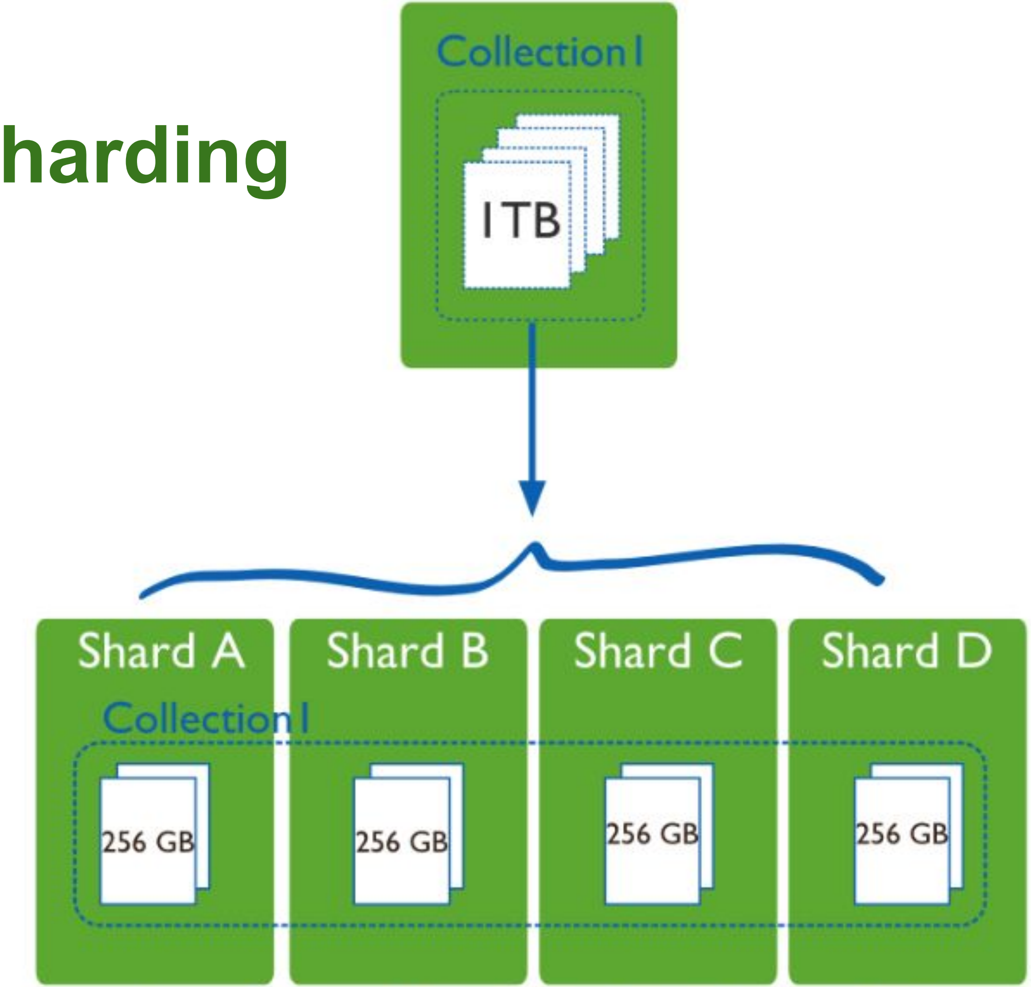
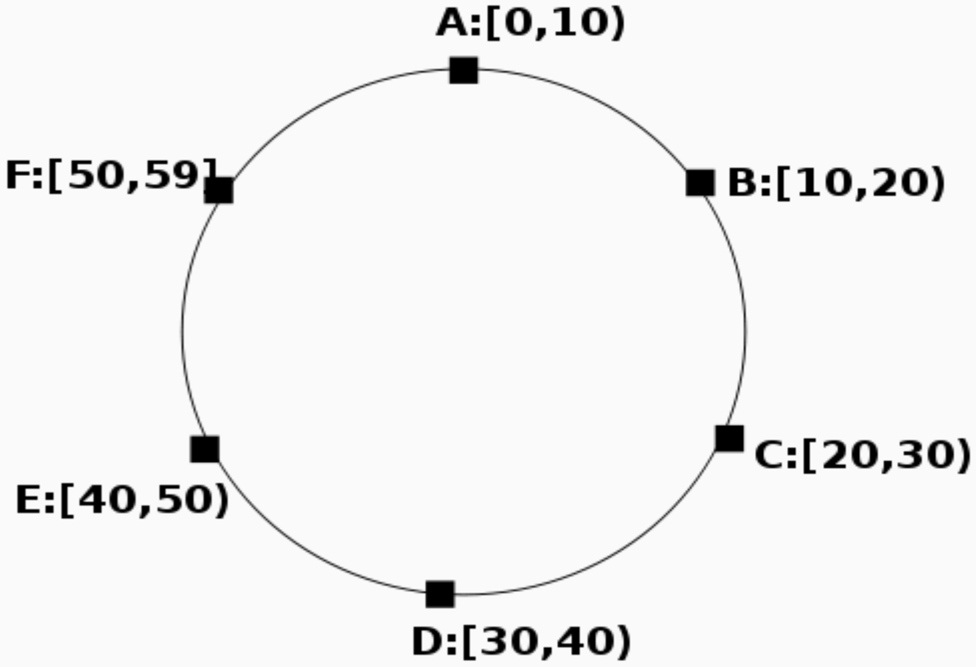


Diagram of a large collection with data distributed across 4 shards.

Data Model Problem

Consistent Hashing then Replicate



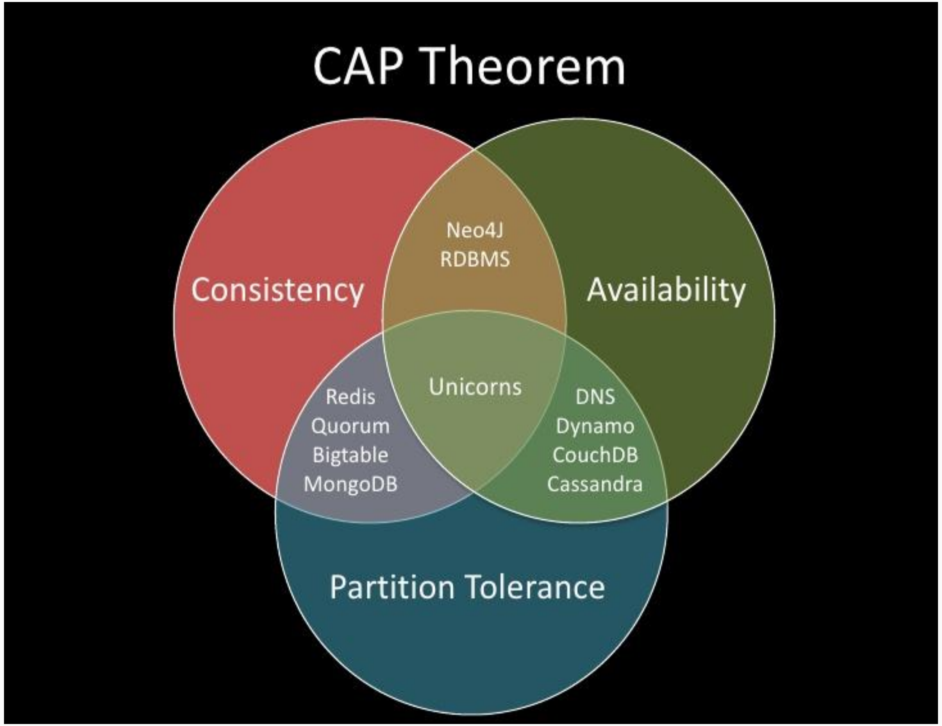
für N=3

Hashwert	Knoten	Replika
3	A	B,C
12	B	C,D
19	B	C,D
20	C	D,E
37	D	E,F
40	E	F,A
54	F	A,B

Consistency

$$R + W > N$$

No. of Nodes Read + No. of Nodes write > Total No. of Replicas



CAP Theorem

Consistency
Availability
Partition Tolerance
Neo4J
RDBMS
Unicorns
Redis
Quorum
Bigtable
MongoDB
DNS
Dynamo
CouchDB
Cassandra

CAP Theorem

- Shared writing project
- Coffee shop closes
- Synchronizing over the phone
- Battery dies

Distributed Computation

MapReduce

Hadoop

- MapReduce API’s
- MapReduce Job Management
- Distributed File Systems(HDFS)
- Enormous Ecosystem

Spark Framework

- Scatter/Gather paradigm (similar to mapreduce)
- More General Data Models(RDDs, Data sets)
- More General Programming Model
- Storage agnostic

Map → Transform

Reduce → Action

Kafka

- Focuses on Real-time analysis, not batch jobs
- Streams and streams only
- Except streams are also tables (sometimes)
- No cluster required !

Messaging

- Means of loosely coupling subsystems
- Messages consumed by subscribers
- Created by one or more Producers
- Organized into topics
- Processed by brokers
- Usually persistent over a short time

Messaging Problem

- What if a topic gets too big for a single computer ?
- What is one computer is not reliable enough ?
- How strongly can we guarantee delivery ?

Lambda Architecture

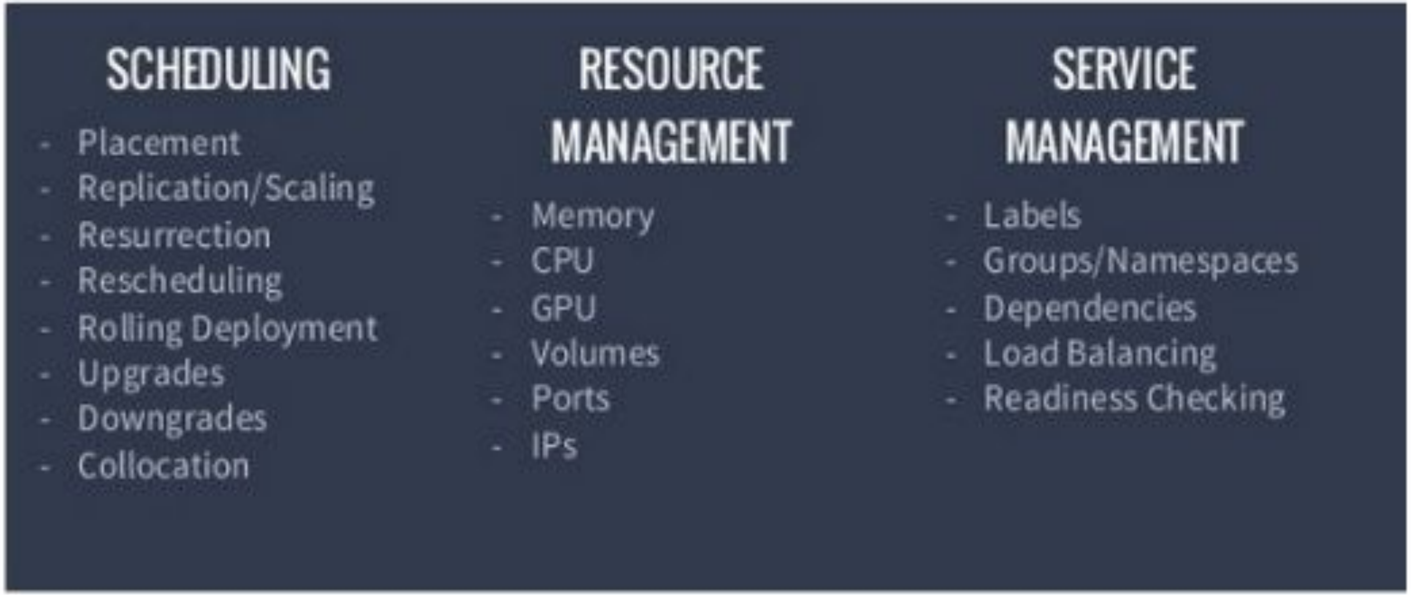
Function as a Service(FaaS)

Event Driven System

Docker Fundamentals

Docker Fundamentals Slides

Container Orchestration duties



SCHEDULING

- Placement
- Replication/Scaling
- Resurrection
- Rescheduling
- Rolling Deployment
- Upgrades
- Downgrades
- Collocation

RESOURCE MANAGEMENT

- Memory
- CPU
- GPU
- Volumes
- Ports
- IPs

SERVICE MANAGEMENT

- Labels
- Groups/Namespaces
- Dependencies
- Load Balancing
- Readiness Checking

Karl di mesosphere: https://www.youtube.com/watch?v=C_u4_I84ED8

Future Scope

- How do people manage deployment of **machine learning** models on Kubernetes ?
- Is docker be used for **self-driving cars** ? Do you deploy one single cluster to manager the entire car ?
- How we can do the distributed IoT deployment ?
- New Business Opportunities…
- And, lots of innovation.

Conclusion

Docker is a fantastic tool for building modern application backends—but it is still just a tool.

If Kubernetes fulfills its mission, it will eventually fade into the background. There will come a time when we talk about Kubernetes like we talk about compilers or operating system kernels. Kubernetes will be lower level plumbing that is not in the purview of the average application developer.

Q/A

Thank you

Ask any further questions.

@mohan08p

@mohan08p

/in/mohan08p

@mohan08p