Tutorial: Running MongoDB and MEAN Apps in Docker and Kubernetes

Vadim Polyakov

About Me

TechSmith – Cloud Services

Prepared Mind – CTO

Inovalon - R&D, Enterprise Architecture, Software Dev, Healthcare

AOL Advertising - High volume transactions, Big Data

3 Startups – Healthcare and Financial Services

Electrical Engineer – Control Systems

FIRST Robotics Mentor

TechSmith®

inovalon™ healthcare empowered

USi An AT&T Company

THOMSON PROMETRIC™

Aol.

meridium

RSS™ RESTRICTED STOCK SYSTEMS

TruCode® empowering coders. delivering results.™

Fundamental Economic Shifts

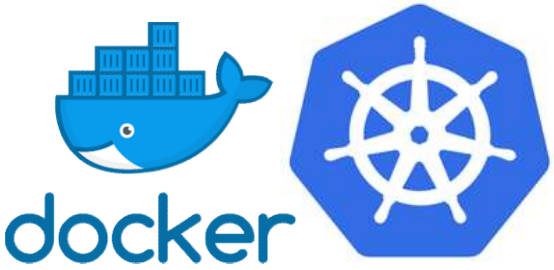
Desktop

Web

Data center

Cloud

Deployment Team + Process



Automated + state

VirtualBox and Vagrant setup

Change directory to USB

```
vagrant box add ./build-mongodb-2017.box --name polyakov/docker-k8s-build –force
```

AWS setup

Launch AMI: ami-bd2c73ab

Detailed instructions: https://github.com/polyakov/kubist

Docker

“Docker containers wrap up a piece of software ... This guarantees that it will always run the same, regardless of the environment it is running in.” (https://www.docker.com/what-docker)

Largely eliminates risk of deployment and rollback

What is a container?

Container is a process, not a VM.

Linux kernel features to isolate processes – namespaces, cgroups, copy-on-write

What is a Docker?

Docker is an opinionated configuration of namespaces, cgroups, iptables, ...

All wrapped up in a nice API and management tools.

Docker

Managed as code – Dockerfile

Simple and fast to build images

Git-like semantics (push, pull, diff…)

Repository of images (public, private)

Management API

Starts in milliseconds (it's a process)

```
1  FROM fedora
2
3  # Install nodejs and npm
4  RUN dnf -y update
5  RUN curl --silent --location https://rpm.nodesource.com/setup_4.x | bash -
6  RUN dnf install -y nodejs
7
8  # Show nodejs and npm versions installed
9  RUN node -v
10 RUN npm -v
11
12 # Set port for nodejs to listen on and expose it
13 ENV PORT 8080
14 EXPOSE 8080
15
16 # Set production environment for nodejs application
17 ENV NODE_ENV=production
18
19 # Make directory for our nodejs project
20 RUN mkdir /app_root
21 RUN mkdir /app_root/app
22 RUN mkdir /app_root/node_modules
23
24 # Inject package.json into directory and install all dependencies required
25 # to be cached in order of making future builds faster
26 ADD ./node_modules /app_root/node_modules
27
28 # Add code of our nodejs project with respect to gitignore
29 ADD ./app  /app_root/app
30
31 # Run it!
32 CMD ["node", "/app_root/app/server/index.js"]
33
```

The plan

- Workshop: Dockerize the application
- Kubernetes intro
- Workshop: deploy application stack to kubernetes
- Workshop: try to break the app

https://github.com/polyakov/kubist

Workshop: Dockerize your application

Goal: Build a Docker application container and run it with MongoDB container.
It's "embarrassingly easy."

Steps

- Create Dockerfile
- Create Docker **image**
- Test Docker **container**
- Publish

Commands

docker info
docker build
docker ps
docker images
docker run
docker login
docker push
docker pull

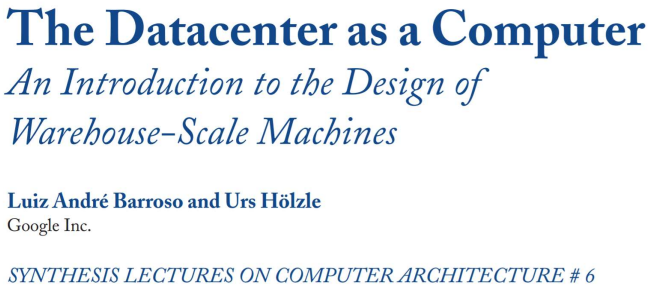
https://github.com/polyakov/kubist
Workshop Steps

Kubernetes

Applications “run on no smaller a unit than clusters of hundreds to thousands of individual servers.”

“Therefore, the machine, the computer, is this large cluster or aggregation of servers itself and needs to be considered as a single computing unit.”

Kubernetes largely eliminates the process of deployment



Kubernetes

Key concepts

- Node
- Pod
- Volume
- Replication Controller
- Service

Kubernetes Components - Node

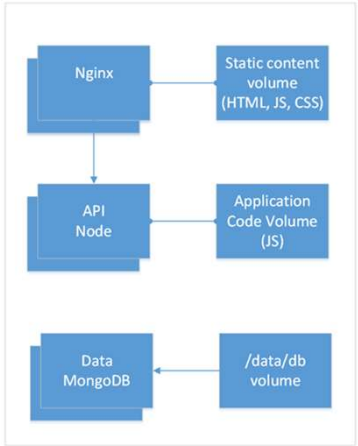
Node - physical resources

Registers with the master

Resource capacity

Labels

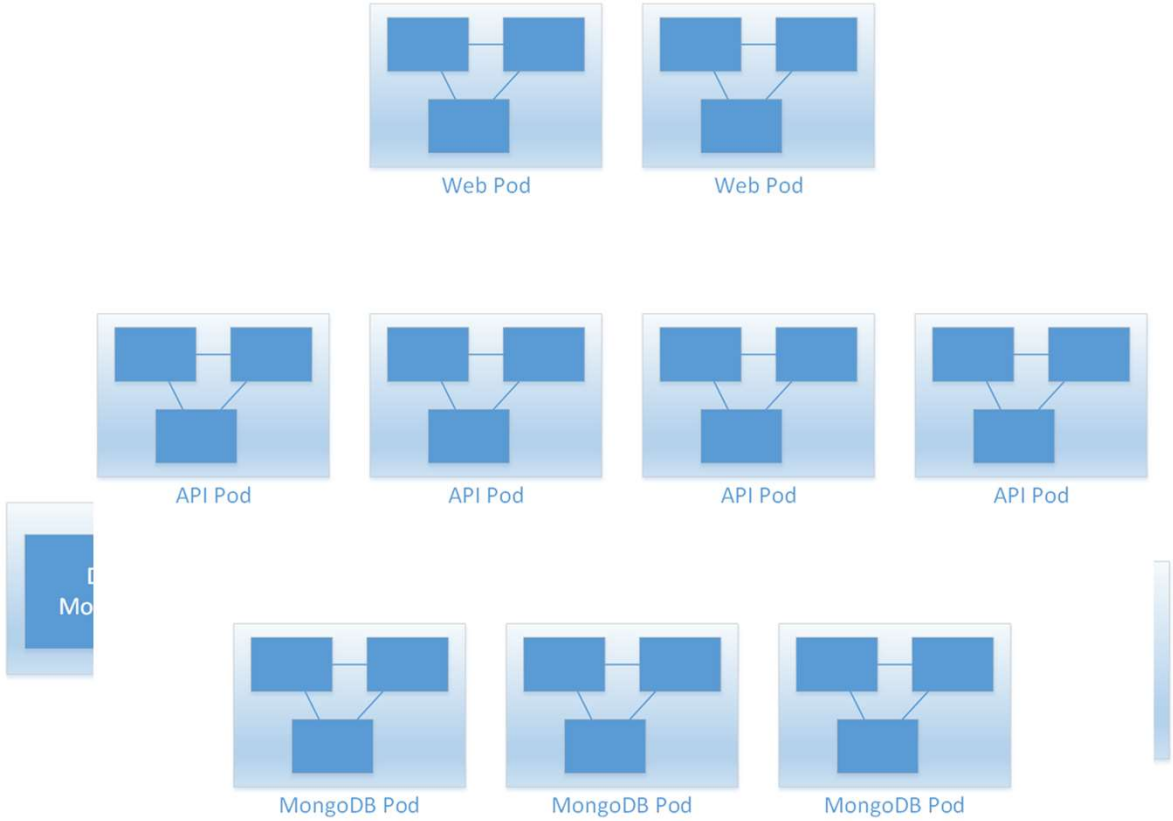
- tags used to filter the environment
  e.g. has kernel config for MongoDB



Kubernetes Components - Pod

Pod

- One or more co-located components with shared context (cgroups, namespaces)



Kubernetes Components - Replication Controller

Replication Controller

- Keep certain number of pod replicas running

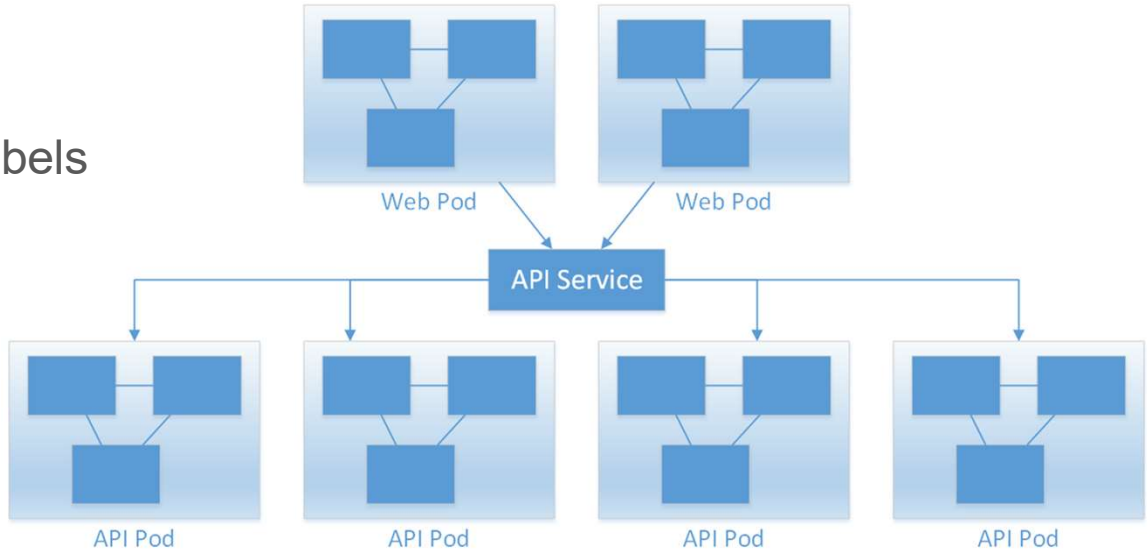
Kubernetes Components - Services

Services

- Set of pods
- Typically selected using labels
- Virtual IP

External Services

Headless Service



Kubernetes Components - Volumes

Container files deleted on restart

Volumes

- Persistent storage across restarts

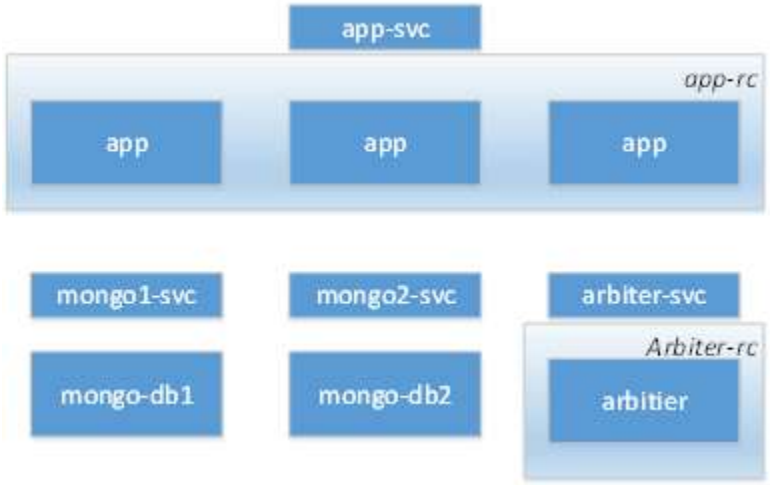
- Shared storage within a pod

Large and growing number of volume types

gitRepo, secret,

- emptyDir
- hostPath
- gcePersistentDisk
- awsElasticBlockStore
- nfs
- iscsi
- fc (fibre channel)
- flocker
- glusterfs
- rbd
- cephfs
- gitRepo
- secret
- persistentVolumeClaim
- downwardAPI
- projected
- azureFileVolume
- azureDisk
- vsphereVolume
- Quobyte
- PortworxVolume
- ScaleIO

App deployment



Deployment as code

API

Service: svc-app.json

Replication Controller: rc-app.json

MongoDB

Services: svc-arb.json
svc-mongo-db1.json
svc-mongo-db2.json

Naked pods: pod-mongo-db1.json
pod-mongo-db2.json

Replication Controller: rc-arb.json

Workshop: Schedule application into the cluster

Deploy scaled application to our cluster

Steps	Commands
- Deploy MongoDB Replica Set	kubectl get
- Deploy Application Tier	kubectl describe
- Expose application as a service	kubectl create
	kubectl

https://github.com/polyakov/kubist
Workshop Steps

MongoDB on Kubernetes

“Datacenter as a Computer”

Most nodes in the cluster are “cattle” not “pets”

Some nodes are special

Databases and DB servers are special

- Working set, kernel config
- Data is an asset

MongoDB on Kubernetes

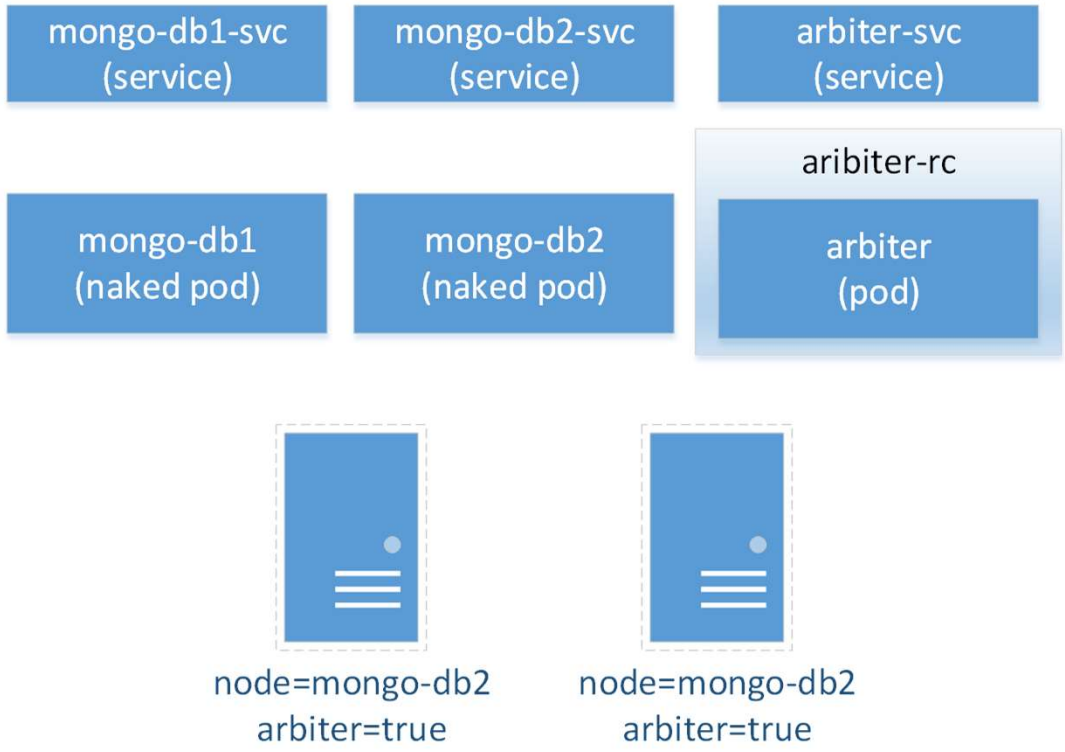
Replication Controller?

- Restart automatically on failure
- Schedule on multiple nodes

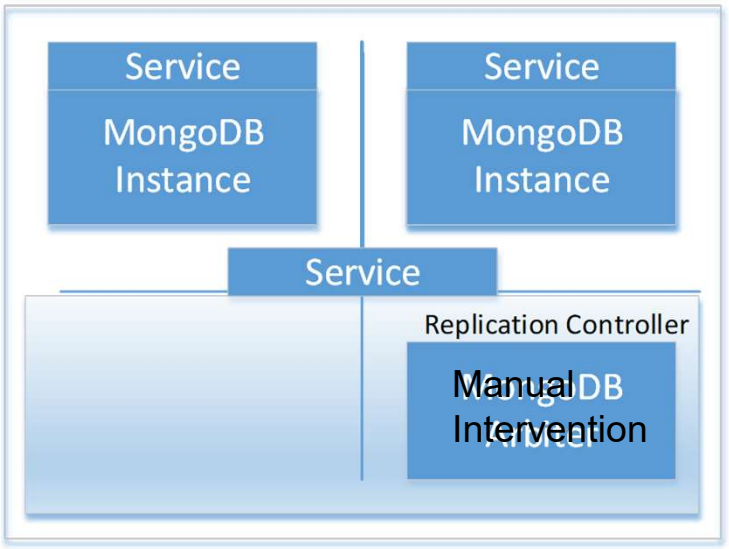
Naked Pods

Services

- Discoverability
- DNS
- Virtual IP



MongoDB on Kubernetes: Demo



Workshop: Self-healing in App and MongoDB tiers

Play with destroying your app

Steps

- Drop an app pod
- Drop a MongoDB pod
- Drop MongoDB node and arbiter
- Drop a server

https://github.com/polyakov/kubist

Resources

Tutorial script and instructions: https://github.com/polyakov/kubist

VirtualBox: https://www.virtualbox.org/

Vagrant: https://www.vagrantup.com/

Kubernetes Blog: http://blog.kubernetes.io/

Thank you

https://www.linkedin.com/in/vadimpolyakov

https://github.com/polyakov