Project Presentation Stage 1

Stefanie Ziltener, Marc Heimgartner, Benjamin Bürgisser, Simon Tännler

Containerized Microservices deployed through Kubernetes

Dummy Image

- Components embedded in (Docker) Containers

- Containers have (ideally) one single responsibility

Frontent and API

- User inputs term
- Click on register
- Send request to API

Dummy Image

Timeseries: MongoDB (I)

- Request Handler stores Term in MongoDB

- Persistency guaranteed by GCE [Persistence Disk](#)

Dummy Image

Twitter Service

- Gets notified of newly created Terms

- Streaming stops and restarts with the new Term added for <u>tracking</u>

- Arriving Tweets are immediately stored into the Queue

Dummy Image

Worker Queue

- <u>Redis</u>: in-memory data structure store

- A FIFO queue of Strings (Tweets)

- Load generator

  - API endpoint to add Strings to Queue directly

Compute Workers

- Running Workers process the Queue:

  - Assign Tweet to Term (filtering)

  - Calculate Sentiment

- Length of the Queue defined the number of Workers

  - Scaled through Kubernetes.

Dummy Image

Timeseries: MongoDB (II)

- Workers store the calculated Sentiment into MongoDB.

Displaying Results

- Rest API gets request

- Collect relevant data from MongoDB

- Browser renders data

- Socket gets opened for continues pushes

Architectural Styles

- Client / Server through Rest API

- Event-Driven notification of Term updates

- Pipes and Filters

- Blackboard: Redis Queue

  - Factory: Twitter Service

  - Worker: Compute Workers

- Highly decoupled

Do you even scale?
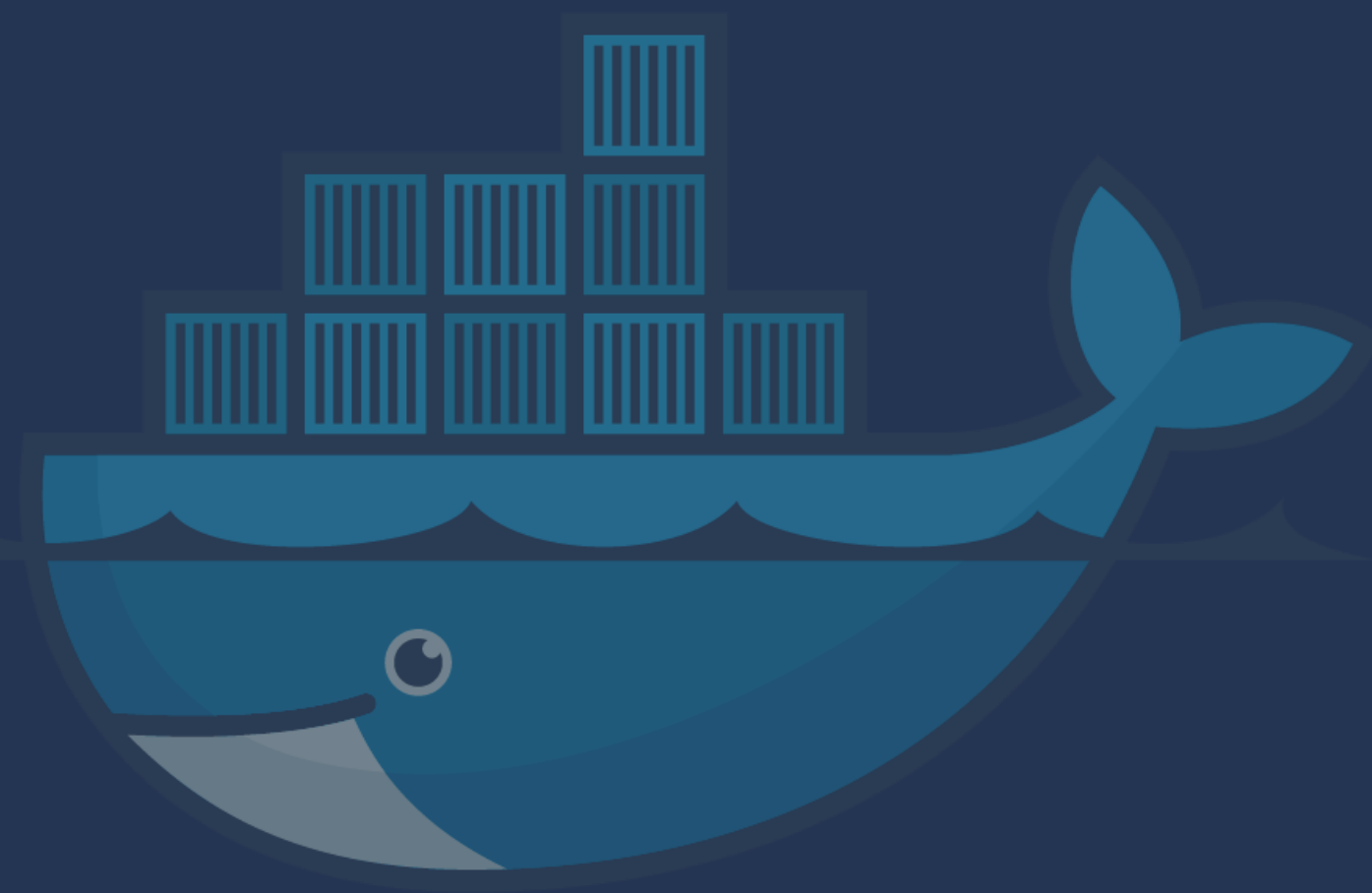
Kubernetes in one slide

@marc chasch ächt du das no mache?

What does <u>Kubernetes</u> do for us

- Every component is potentially scalable through Kubernetes

    - Even <u>MongoDB</u>!

- Fault Tolerance:

    - Container recovery through Kubernetes

    - Decoupled design and Microservice

- Elasticity

    - Container scaling through Kubernetes

Concernes

- High lock-in to Kubernetes

- Twitter

  - Only 400 Term, thus no scaling

  - May not match to terms perfectly

- Redis: may become a bottleneck

  - but we highly doubt it

-

Technology Zoo - Platform

- Cloud Platform: Google Container Engine (GCE)

  - Easy support of Kubernetes

- Containerization: Docker

  - Popular Container engine

- Container orchestration: Kubernetes

  - Popular Container orchestration

Technology Zoo - Backend

- Programming Language: Google Go

    - New Programming language 🎉

    - Uniquely suited for Web development

    - Have I mentioned it's fast?

- Terms Storage: MongoDB

    - Easy data schema

- Queue Storage: Redis

Technology Zoo - Frontend

- Frontend: <u>ViewerJS</u>

  - Similar to Angular

- Webserver: <u>nginx</u>

  - Battle-proven Webserver

Dev Environment

Docker containers using Docker Compose

Demo

Questions?