DAAD Summer School 2019

ulm university universität ulm

Tools & Concepts for Cloud Deployments

Exercise 7: Kubernetes; Reliability & Availability

Christopher B. Hauser

Institute of Information Resource Management, Ulm University

SummerSchool 2019, Curitiba

Section 1

Exercise 7: Kubernetes; Reliability & Availability

Overview

Welcome to exercise 7. This time we will have the following lessons:

1. Container Orchestration with Kubernetes
2. Concept of Reliability and Availability

Lessons learned

Section 2

Answers to questions

Lesson 1: Container Orchestration with Kubernetes

Questions: Kubernetes

What Featuers does Kubernetes provide?

- Automatic binpacking
- Self-healing
- Horizontal scaling
- Service discovery and load balancing
- Automated rollouts and rollbacks
- Secret and configuration management
- Storage orchestration
- Batch execution

Where in our Cloud Stack do you place Kubernetes?

Cloud Stack	Example	Deployment Tool
Application Component	Mediawiki	Dockerfile/Bash
Containers	Docker	Kubernetes
Virtual Resource	Instance m1.small	-

Lesson 2: Concept of Reliability and Availability

Questions: Consistency Guarantees

What is ACID? What is eventual consistency?

ACID: Atomicity, Consistency, Isolation, Durability - describes that transactions with various changes or queries to a data store runs as one atomic action, while the database guarantees strict consistency, parallel running actions don't harm each other by isolating them, and that written changes are durably persisted and never lost. These guarantees are very strict and are implemented by locking mechanisms, which decreases availability and performance.

Eventual consistency - describes that changes will be visible at an unspecified point in time in the future to other queries. This guarantee is in contrast to ACID, but allows an increase on availability and performance.

What are the three categories of CAP?

Concistency, **A**vailability, **P**artition tolerant - and only two (but never three) of these three features can be achieved

Questions: Distributed Database Systems

What is the difference between replication and sharding/partitioning?

Replication copies data on other locations, while sharding splits the data in pieces and distributes them to locations. Distributed NoSQL database systems often use both at the same time, while relational database systems (e.g. MySQL) often don't shard the data but allow replication.

How does replication relate to the CAP theorem?

Replication can improve the availability property and can be used to recover in case of failures or partitions. Replication hence has direct influence to the AP properties.

How does sharding relate to the CAP theorem?

Sharding distributes the workload to many nodes in parallel, and hence influences the performance (availability). In case of failures or partitions not 100% of