images/thi_logo.pdf

Technische Hochschule Ingolstadt

Specialist area Computer Science
Bachelor’s course Computer Science

Bachelor’s thesis

Subject: Conception, Implementation, and Evaluation of a Highly Scalable and Highly Available Kubernetes-Based SaaS Platform on Kubernetes Control Plane (KCP)

Name and Surname: David Linhardt

Issued on: TODO: Insert Issue Date

Submitted on: TODO: Insert Submit Date

First examiner: Prof. Dr. Bernd Hafenrichter

Second examiner: Prof. Dr. Ludwig Lausser

Abstract

Contents

Glossary

1 Introduction

1.1 Problem Statement and Motivation

1.2 Objectives and Scope

1.3 Structure of the Thesis

2 Fundamentals

2.1 Kubernetes and Multi-Tenancy

Introduction and Motivation As the de facto standard for deploying and managing *cloud-native applications*, Kubernetes plays a pivotal role in modern cloud architecture . Kubernetes works as an application orchestrator for *containerized, cloud-native microservice* apps, meaning it can deploy applications and dynamically respond to changes . It offers a platform for declarative configuration and automation for containerized workloads, enabling organizations to run distributed applications and services at scale .

Multi-tenancy plays a fundamental role in modern cloud computing. By allowing multiple tenants to share the same infrastructure through virtualization, it significantly increases resource utilization, reduces operational costs, and enables essential features such as VM mobility and dynamic resource allocation . These benefits are critical for cloud providers, as they make the cloud business model economically viable and scalable.

However, while multi-tenancy is indispensable for efficiency and cost-effectiveness, it simultaneously introduces complex security challenges, especially in shared environments where resource isolation is limited. As such, understanding and addressing multi-tenancy is essential when designing and securing modern cloud-native platforms .

Overview of Kubernetes Kubernetes was originally developed at Google and released as open source in 2014 . *Containerization* is a way to bundle an application's code with all its dependencies to run on any infrastructure thus enhancing portability . This comes with additional advantages that can be leveraged by Kubernetes, including vertical and horizontal autoscaling facilitated by quick container boot times, along with self-healing mechanisms and support for distributed, resilient infrastructures .

Kubernetes Resource Isolation Mechanisms

Multi-Tenancy Challenges in Kubernetes

Approaches to Multi-Tenancy in Kubernetes

Relevance to SaaS and this Thesis

2.2 Kubernetes Control Plane (KCP)

2.3 SaaS Architecture and Automation

3 State of the Art and Related Work

3.1 Zero-Downtime Deployment Strategies

3.2 Kubernetes Scaling Methods

3.3 Multi-Tenancy Concepts in the Cloud

4 Conceptual Design

4.1 System Requirements

4.2 Architecture Design with KCP for SaaS

4.3 Automated Deployment Strategies

5 Prototypical Implementation

5.1 Infrastructure with KCP

5.2 Tenant Provisioning (Automation, Multi-Tenancy)

5.3 Scaling Mechanisms (Horizontal Pod Autoscaler)

5.4 Monitoring and Logging (Prometheus, Grafana)

6 Evaluation

6.1 Performance Measurements (Downtime, Latency, Scaling)

6.2 Scaling Scenarios & Optimizations

6.3 Discussion of Results

6.4 Related Work

7 Conclusion and Outlook

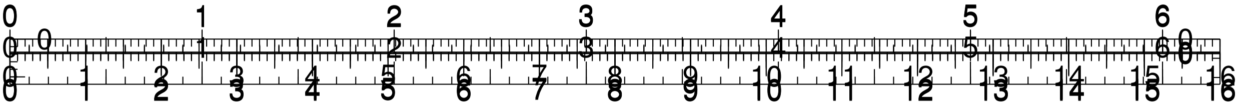
7.1 Summary

7.2 Personal Conclusion

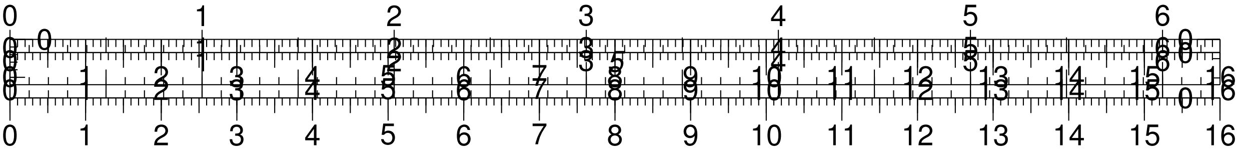
7.3 Future Outlook

References

List of Figures


0 1 2 3 4 5 6
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Appendix


0 1 2 3 4 5 6
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16