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: TODO: Insert Second examiner

Abstract

Contents

1 Introduction 4

1.1 Problem Statement and Motivation . . . 4

1.2 Objectives and Scope . . . 4

1.3 Structure of the Thesis . . . 4

2 Fundamentals 4

2.1 Kubernetes and Multi-Tenancy . . . 4

2.2 Kubernetes Control Plane (KCP) . . . 4

2.3 SaaS Architecture and Automation . . . 4

3 State of the Art and Related Work 4

3.1 Zero-Downtime Deployment Strategies . . . 4

3.2 Kubernetes Scaling Methods . . . 4

3.3 Multi-Tenancy Concepts in the Cloud . . . 4

4 Conceptual Design 4

4.1 System Requirements . . . 4

4.2 Architecture Design with KCP for SaaS . . . 4

4.3 Automated Deployment Strategies . . . 4

5 Prototypical Implementation 4

5.1 Infrastructure with KCP . . . 4

5.2 Tenant Provisioning . . . 4

5.3 Scaling Mechanisms . . . 4

5.4 Monitoring and Logging . . . 4

6 Evaluation 4

6.1 Performance Measurements . . . 4

6.2 Scaling Scenarios & Optimizations . . . 4

6.3 Discussion of Results . . . 4

6.4 Related Work . . . 4

7 Conclusion and Outlook 4

7.1 Summary . . . 4

7.2 Personal Conclusion . . . 4

7.3 Future Outlook . . . 4

References 4

List of Figures 4

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

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

Appendix