Şükrü Karadağ

DevOps Engineer

sukru.karadag28@gmail.com | 905458006475 | linkedin.com/in/sukru-karadag

medium.com/@sukrukaradag | github.com/SukruKaradag | İzmir, Turkey

PROFILE

Having a strong everyday-learning mentality combined with open-mindedness and determination to learn new skill sets and adapt them to new environments faster. Thanks to the working experiences in multicultural environments and being a part of mullticultural/multifunctional IT teams with complete coherence, now adaptable to changing working environments at ease. Having conducted Agile methodologies integrated with CI/CD Practices.

SKILLS

Python | AWS | Git-Github | Linux | Bash Scripting | Docker | Kubernetes | Helm | Jenkins

AWS Cloud Formation | Ansible | Terraform | Jira | Slack | Prometheus | Grafana | CI/CD

Windows Server | HTML5 | CSS | Yaml | Json | Groovy | Agile | Nginx | Linux System Administration

PROFESSIONAL EXPERIENCE

Jan 2022 – Aug 2022

DevOps Engineer, *Rainpark Solutions*

- In my recent project, I worked mainly on a complete CI/CD Pipeline for React in frontend, Spring Boot java Application in backend and PostgreSQL Database
- Dockerized Web Application, developed the frontend with React and backend with the Spring Framework, using Spring
- MVC as the web framework and PostgreSQL as the database CI/CD Pipelines, and deployment on a Kubernetes cluster with monitoring.
- Kubernetes cluster was created and managed with Rancher. Jenkins was used as the CI/CD automation tool and we created all the infrastructure on AWS EC2 Service.
- The code was developed in Java and Maven was used as the build tool. So I used Maven Wrapper for the testing, packaging, and installing phases. I spun up the development server through a CloudFormation template.
- We used some tools like Compose and Customize tools. We converted the Docker-compose files to Kubernetes definition files by using the Compose tool. We also used the Kustomize tool to add some customization to these definition files -like changing replica numbers or image tags.
- We also used Rancher for creating, controlling, and monitoring Kubernetes clusters for staging and production environments. We used Jenkins pipeline scripts to deploy the application to both staging and production environments.
- I used Rancher to create and manage our Kubernetes clusters. To install the Rancher, I used the Helm chart. With Rancher, we easily made changes in the cluster via its dashboard, add nodes, delete nodes, edit configuration files, and used kubectl on its terminal.
- I used as monitoring tools: we monitored the applications in the cluster with Prometheus and Grafana.

EDUCATION

2021 – present

Bachelor, *Istanbul University*
Management Information Systems

2016 – present

Bachelor, *Ege University*
History

CERTIFICATES

Turkcell
Linux 101, 102, 103 and 104 Certificates

Amazon Web Services (AWS
AWS Cloud Practitioner Essentials

Linkedin Learning
Learning Kubernetes, Learning Docker, Learning Jenkins, Learning Terraform, Learning Bash Scripting, Become a Linux System Administrator, Lean and Agile, Learning Django, Learning Python

Cisco Networking Academy
| Networking Essentials Certificate

Özgür ÖZTÜRK's Udemy Courses
A'dan Z'ye Docker Certificate, Kubernetes Temelleri Certificate, Bulut Bilişim Temelleri ve AWS Çözüm Mimarlığı

Bilgeİş - ODTÜ
Python Programming - 2

LANGUAGES

Turkish
Native

English
C1