# Matthew Clegg - Kubernetes Administrator

m@cle.gg
+44 (0) 7478 575533

Over **17 years** of industry experience, developing with client, server and infrastructure technologies. Skilled in automation cloud orchestration tools (Kubernetes, Terraform, Helm, Ansible etc), server-side technologies (GoLang, Python, PHP) & client front-end languages (HTML, CSS and JavaScript). I have experience creating and hosting websites using complex math & basic logic for most common content management systems. My previous work experience includes projects with numerous digital agencies, startups and corporations.

---

## Senior Site Reliability Engineer

**Old Street, London 2022 - present**

YAPILY

Designed & built a cross-regional disaster recovery capability in GCP by: bringing all resources under Terraform management, automating/consolidating of standalone scripts, building a passive cluster in a secondary region and cleanly deploying applications to the new region. Designed, developed, and implemented software integrations based on user feedback. Troubleshooted production issues and coordinated with the development team to streamline code deployment. Applied cloud computing skills to deploy upgrades and fixes. Implemented automation tools and frameworks (CI/CD pipelines). Automated re-encryption process of large amounts of sensitive data. Conducted systems tests for security, performance, and availability. Analysed code and communicate detailed reviews to development teams to ensure a marked improvement in applications and the timely completion of projects. Optimised the company's computing architecture. Assisted in acquiring information security standards ISO27001. Made Open Source Contributions to update the Terraform provider for Mozilla SOPS to write files using KMS. I have provided multiple training sessions for all of engineering. One on Rebasing practices in Git and another on migration of GCP resources into Terraform.

cicd devsecops docker gcp python shell terraform web

---

## VP Site Reliability Engineer

**Wood Lane, London 2020 - 2022**

Responsible for automating everything from A to Z using Terraform, AWS, Azure, etc. Ensuring 'the lights stay on', to deliver the best possible experience to customers. Migrated web services into Docker/Kubernetes from ElasticBeanstalk & migrated from BitBucket to GitLab. Proactively monitored the health of the platform, to detect and resolve threats, defects and outages before they had any meaningful impact on users. Enhanced teams in regards to communication, collaboration and integration between software developers and infrastructure stakeholders in the software delivery lifecycle. Worked closely with product stakeholders, such as designers and product managers to gather requirements and refine problems into focused deliverables. Advised with a pragmatic approach to operation/product teams on topics such as automation, deployments and architecture. Provided training sessions for keen developers to attend regarding Docker & Kubernetes. Responsible for supporting a team of SREs in monitoring the platform using logs, metrics, tracing, amongst other observability. I aimed to improve their existing platform concerning scalability, deployments and reliability so they require less maintenance in the future. I used my start-up mentality to be responsible, energetic, ambitious, adaptable and willing to jump in wherever might be needed.

aws azure devsecops docker go-lang people-are-important php python shell terraform web

---

## SRE / DevOps Engineer

**Pullach, Bavaria 2019 – 2020**

SIXT

Initially configured EKS Clusters via TerraForm to be provisioned from GitlabCI pipeline. I created a Jenkins pipeline framework using Groovy scripts for dockerizing Spring Boot application microservices. I created a framework for deploying a Kafka proxy/zookeeper for creating ephemeral microservice preview environments. I configured API Gateway with TerraForm to run a lambda for deploying generated microservice Kubernetes manifests. I implemented a Prometheus based monitoring solution and synced observability requirements of the Orange Platform.

aws docker

---

## Site Reliability Engineer

**Munich, Germany 2018 – 2020**

Responsible for building and maintaining cloud based infrastructure for SAAS platform used by SME customers. Responsible for automating everything from A to Z using Terraform, Ansible, AWS lambda functions, etc. I was responsible for supporting a team of SREs in monitoring the platform using logs, metrics, tracing, amongst other observability stacks such as ELK and NewRelic. I advised product teams on topics such as automation and architecture. I aimed to enhance teams in regards to communication, collaboration and integration between software developers and infrastructure stakeholders in the software delivery lifecycle. While at Personio I migrated their web services into Kubernetes in AWS using GitLab. I created and managed a GitLab service in an automated fashion. I used a GitLab instance to provision Kubernetes environments in AWS. I migrated all microservices to Kubernetes services in AWS environment using direct connect. I aimed to improve their existing platform concerning scalability, deployments and reliability so they require less maintenance in the future. I supported the internal team on operations topics regarding deployment & automation. I provided training sessions for keen developers to attend regarding Docker & Kubernetes.

aws cicd devsecops docker php python shell terraform web

---

## Senior Technical Operations

**Munich, Germany 2016 – 2018**

During my time at Freeletics I was responsible for running their heavily used platform to sustain the api/web services 24x7x365. I completed this by monitoring the platform to know of issues before users and proactively responding to maintain the system. While at Freeletics, I introduced KOPS (Kubernetes Operations) to manage Kubernetes. Previously, the cluster had been created 'by hand' to use a combination of flannel/AWS route tables to manage internal networking. I improved the cluster design by; enabling load distribution across multiple availability zones in AWS, replacing a hardcoded ingress service with a dynamic ingress controller, fixing several security issues, reducing operating costs and, most importantly, provided the ability to autoscale the number of servers required by the cluster based on the current demand. The launch script for the cluster was also documented as a repeatable script (that ran inside docker) that could create a new Kubernetes cluster by; calculating available subnet CIDRs in a VPC, generating required keys/certificates, updating DNS records, defining all required resources in terraform, setup networking using Weave and initializing a helm tiller to await for future deployments. I created a helm chart that described all micro-services to be done as a single deployment . This improved the workflow by ensuring that the same versions and configuration settings (routing, Memory/CPU, etc) would be deployed on all production/staging/QA environments. It also enabled the teams to use HTTPS & subsequently HTTP2 for all environments.

aws devsecops docker people-are-important python shell web

---

### Hobbies & interests

My favourite hobbies include long distance cycling, short distance running and hiking.
I am also interested in cooking and photography.