Mariahilfer Straße 126/25
1070 Vienna
+43 670 6060886
simon.tesar@gmail.com
www.simt.io

Simon Tesar

Employment History

Since 2019 **Backend Engineer**, *Unwired Networks*, Vienna.

At Unwired Networks I developed several internal and public-facing services in a team or single-handedly:

- Internal remote device management system distributing configuration, collecting metrics and serving as SSH-jumphost
- OpenVPN-server management client and API via GRPC
- MQTT-based device log system with Elasticsearch backend
- Reimplementation of legacy APIs based on GraphQL

Besides my function as a software engineer I held responsibility for production infrastructure:

- Managed GKE clusters, bare-metal servers and corresponding accessories like monitoring
- Replaced static ingress and logging setups with dynamic solutions
- Moved existing infrastructure to a manageable state via a combination of Ansible and Kustomize

Primary technologies: Go, Kubernetes, Kustomize, GCP, AWS, Ansible, Docker, MySQL, NodeJS

2018–2019 **Site Reliability Engineer Contractor**, *Cisco Systems Austria*, Vienna.

At Cisco Systems I was part of a globally distributed team of Site Reliability Engineers. Besides the project listed later in this document we were each assigned to a team of software engineers and responsible for consulting them on developing cloud-native applications and aiding with deploying on our managed infrastructure. This mostly consisted of writing things like Helm charts, custom Terraform providers and Ansible playbooks.

Primary technologies: Go, Kubernetes, Helm, Kustomize, AWS, Terraform, Ansible, Terraform, Docker, Python, VMware vSphere, Github Actions

2015–2017 **DevOps Engineer**, *APA-IT Informations Technologie*, Vienna.

My old employer Gentics Software was merged into its parent company APA-IT. As most of the physical infrastructure was now managed by our hosting team I could focus on my work as the DevOps engineer of the team. Besides the projects listed later in this document this included responsibility for how we deployed and managed our services and all other infrastructure-related issues and tasks.
I also led workshops and trainings for internal and external employees on cloud-native techologies and our own products.

Primary technologies: Linux, Docker, Kubernetes, Go, Jenkins, GitLab, Traefik, Helm, Ansible, GCP, Openstack, ZFS, MySQL

2013–2015 **Linux System Administrator**, *Gentics Software*, Vienna.

At Gentics Software I was responsible for all server and networking infrastructure. Besides the migration project mentioned later in this document, I designed and implemented new infrastructure concept, held responsibility for the infrastructure part in all customer projects and enhanced our product's development process, deployment, building and packaging.

Primary technologies: Linux, Puppet, Vagrant, VMWare, LXC, Ruby, Jenkins, Nginx, Apache, PHP, Tomcat, MySQL, Java, ZFS

Technology Experience

≥5 Years Go, Linux, DevOps, CI/CD, Docker, Kubernetes, Ansible, Nginx, Apache, MySQL, Debian, Ubuntu, CentOS, RedHat, ZFS, Vagrant, Bash

≥3 Years	GCP, Terraform, Traefik, Helm, Kustomize, Prometheus, Elasticsearch, Kibana, Logstash, Grafana, Jenkins, OpenVPN, Puppet, Chef, VMWare, LXC, FreeBSD, Tomcat
≥1 Year	Github Actions, Python, NodeJS, Java, AWS, GitLab, Vault, GRPC, OpenWRT, MQTT, OracleDB, MacOS

Education

2011–2012 **Technical University of Vienna**, *Software Engineering*.

2006–2011 **HTBLVA Spengergasse**, *IT, speciality in networking technologies*.

Project History

2021 **Bare-metal clusters in a datacenter environment**, *For Unwired Networks*, Vienna.

The project's goal was to create a cluster design focusing on low maintenance and overhead in a datacenter environment. Additionally exiting network traffic needed to be controlled on `Pod`-level without depending on the `Node`'s IP addresses. I used the k3s Kubernetes distribution on bare-metal installations of the k3os operating system to not depend on any traditional virtualization environments and manage upgrades via the included operator. To solve the networking requirements I settled on the Calico project[0] to provision a configured subnet for `Pod` IP addresses and route traffic directly via BGP. As a result each `Pod` was reachable via its own public IP and had its egress traffic originate from it.

Primary technologies: Kubernetes, BGP, Calico, Ansible, Kustomize, general networking and routing

2018 **Openstack, VMWare and AWS cluster management**, *For Cisco Systems Austria*, Vienna.

As a team we developed a framework to create and manage Kubernetes clusters on Openstack, VMWare and AWS in a platform agnostic way. The goal was to provide the other teams of our organization with the necessary tools to manage their own infrastructure on a per-project basis using either public or private cloud environments without resorting specific implementations each time.
The minimum viable product was developed using Python, Ansible and Shellscripts. Certain dependencies like Terraform were extended with custom plugins in Go, a Operator to manage Openstack Snapshots was developed too.
The MVP was used by three teams to manage their development and staging environments while preparing their production environments for transition to Kubernetes. For the final version we decided port the whole framework to Go, unfortunately our parent organization was dissolved and the project never reached completion.

Primary technologies: Kubernetes, Go, AWS, Helm, Terraform, Ansible, Python

2017 **Backend for SaaS platform**, *For Gentics Software*, Vienna.

A software-as-a-service platform for a JVM-based content management system running on Google Kubernetes Engine. A backend service exposedl a REST API via which instances of the software could be managed through their whole lifecycle. The frontend used this service to provide customers a web interface where they could create, configure and modify their own instance of our product.
Although back in 2017 custom resources and the operator pattern were not fully established yet, the sevice essentially mimiced their behaviour. For each instance of the product an internal controller would create the necessary Kubernetes resources to run and expose a `Pod` and track them via custom annotations.

Primary technologies: Go, Kubernetes, GCP, Helm

2016 **CI/CD Platform**, *API-IT Informations Technologie*, Vienna.

I designed, installed and managed a continuous integration and delivery platform for our team of about twenty web and backend developers. The platform was based on Ubuntu, Kubernetes, Ansible, Jenkins, GitLab and our own bare-metal datacenter infrastructure. The projects using the platform were mainly written in Java or PHP besides a few internal services written by myself.

Primary technologies: Kubernetes, Ansible, Jenkins, GitLab

2016 **ZFS storage provisioner**, *For Gentics Software*, Vienna.

An in-cluster service extending Kubernetes with a custom `StorageClass` for the ZFS filesystem. Creating a `PersistentVolumeClaim` resulted in automatic creation of a ZFS dataset which then was exposed via iSCSI or NFS. The service fully supported all available access modes including `ReadWriteMany` and capacity via ZFS quotas.
The provisioner was used for the internal CI/CD platform and the SaaS project included in this project history.

Primary technologies: Go, Kubernetes, ZFS, GCP, Terraform

2013 **Puppet Migration**, *Gentics Software*, Vienna.

I migrated all existing infrastructure consisting of about 30 physical and virtual servers to a better manageable, code-based setup via Puppet. This included managing a central Puppet master and writing Puppet modules for the company's own software.
Two years later I was again responsible for moving this platform into the parent company's Puppet-based management system.

Primary technologies: Linux, Puppet, Ruby, Bash