

Azer Abdullaev

Systems administrator / Site Reliability Engineer

- **Phone:** +49 (176) 772 513 60
 - **E-mail:** azer.abdullaev.berlin+cv@gmail.com
 - **Github:** <https://github.com/reactorcoremeltdown>
 - **Location:** Berlin, Germany
-

Summary

I am a **Site Reliability Engineer** with systems administration background. I am currently focused on and passionate about automating software building pipelines, log and data flows, monitoring and incident reporting, reducing mean time to recovery and improving overall platform resiliency.

Work experience

- 01.2022-01.2024
 - **Software Engineer (full-time)** at **Wayfair GmbH**

Maintaining and enhancing company-wide Istio service mesh installation, participating in on-call shifts, co-working with Kubernetes team on developing and maintaining a repository of Helm packages, developing low-level network connectivity layers based on Envoy and compatible with Istio.

My primary task at Wayfair was developing a dedicated control plane for Envoy proxy that allowed the company to interconnect software deployments across different GCP projects and locations. The control plane also included automatic zero-configuration service discovery for workloads running in CloudRun, a serverless platform of Google, and also GKE deployments. The control plane, and therefore the Gateway, ensured compatibility and interoperability with existing services in the shared project, running on Istio.

- 07.2021
 - **DevOps Consultant** at **MLQRMENU**

Major Infrastructure overhaul, introducing Drone.io as CI/CD platform integrated with Github, introducing k3s as a container orchestration platform, developing a release pipeline, introducing blackbox monitoring, documenting the changes.

The whole deploy pipeline of MLQRMENU was transitioned from Git-hooks based mechanism to GitOps-enabled and cloud-ready platform within just three days.

- 01.2017-01.2022
 - **Site Reliability Engineer (full-time)** at **Pepper Media Holding GmbH**

At Pepper I led the transition of both on-premise and cloud-based infrastructure from old Puppet 3 + Ansible IaC layout to pure Puppet 5 configuration management. This transition allowed us to bootstrap machines and propagate changes faster. During migration we also introduced Hiera, a native Puppet key-value storage that allowed us to separate data from logic.

At the time of transition to AWS ECS we faced another challenge: the log ingestion and processing infrastructure, based on ELK stack, was not ready for dynamic environment. I proposed to use then-experimental Syslog input of Filebeat and direct the JSON-formatted output of NGINX running inside Docker CE directly there. This move allowed us to eliminate grok parsers of Logstash filters for the native NGINX log format, therefore saving computing power and eliminating the need to maintain complex filters.

- 04.2014-01.2017
 - **Systems administrator (full-time)** at **Yandex LLC**

During my time as a systems administrator at Yandex we have established internal processes and runbooks, allowing newcoming teammates to onboard faster. Later on we have significantly reduced our mean time to enable new clusters and services, as well as migrating existing ones, by retrofitting internally-developed monitoring API into well-known tools like Ansible through plugins. For the rest of infrastructure the Salt stack was selected as the best fit.

After several internal organisational restructurings, the systems administration team has been transitioned & therefore embedded into software development team. My task was to migrate an on-premise based Django application and fit it into the new dynamic container-based cloud platform, also internally developed. During migration we found out that on-premise systems mostly relied on static configuration. As a reaction to this I have developed a dynamic healthcheck for HAProxy that allowed zero-config and zero-deploy failover for our MySQL cluster. The solution was tested and deployed minutes before my resignation. Reports say it still works without a single flaw up to this day.

- 08.2012-04.2014
 - **Duty Engineer (full-time)** at **Yandex LLC**

Maintaining hardware in data-centers, replacing broken parts, establishing physical network communications. This was my only time during the whole career where I encountered real hardware on which systems run. At Yandex's DCs, besides maintenance, we also ensured the enablement of hardware, including pre-configuration through remote interfaces(IPMI).

- 03.2012-06.2012
 - **Duty Engineer (full-time)** at **Oversun Scalaxy**

Supporting in-house developed cloud computing platform, responding to incidents, communicating with customers. At Scalaxy we supported a large OpenVZ-based cluster of thousands of virtual machines. During pre-containers and pre-SaaS era we had to invent our own approaches to manage and maintain a fleet of machines. Most of the tooling used for on-premise static deployments, such as Chef for IaC and Zabbix for monitoring, had to be retrofitted into dynamic environment.

Knowledge & Tech Skills

- **Service Mesh**
 - Fresh experience with *Istio* service mesh and compatible solutions.
- **Container runtimes and orchestration**
 - Good knowledge of *Kubernetes* and its APIs; had also experience with compatible platforms, like *K3s* for edge computing, and also low-level container orchestration components, like *Docker*, *Docker registry*, *Podman*, and *Systemd*. Good understanding of OS composition and integration.
- **Cloud platforms**
 - Fresh experience with *GCP*, mostly *Compute* and *GKE* components. Had also worked previously with many components of *AWS*, such as *ECS*, *Elastic LoadBalancers*, *RDS*, *EC2*, *S3*, and others.
- **CI/CD and code hosting platforms**
 - Strong knowledge of *CI/CD* platforms. Fresh experience with *Buildkite*, *ArgoCD* and *Github Actions*; running *Drone.io* for personal infrastructure on a daily basis; previous experience with *Jenkins* and *Gitlab CE*.
- **VCS**
 - Strong knowledge of *Git*: familiarity with the *GitOps* concept, ability to build software development flows on top of *Git*, as well as understanding of full software development cycle.
- **Infrastructure as Code**
 - Strong knowledge and understanding of *IaC* concept. Fresh experience with *Puppet*(and *Hiera*) and *Terraform*; previous experience with *Ansible* and *Salt*.
- **Logging & data flow**
 - Good knowledge of log collecting and aggregating systems. Had worked with *ELK* stack, developing an entire pipeline of log delivery and processing for a business.
- **Web servers & load balancing**
 - Good knowledge of load balancing software, as well as L4 and L7 proxying. Fresh experience with *Envoy* and *NGINX*. Had also worked with *HAProxy* for web and database load balancing.
- **Programming languages**
 - Good knowledge of *Go* programming language: developed command line utilities, as well as *Kubernetes operators* and *control planes*; advanced knowledge of *Bash scripting* for batch processing, systems configuration, and software building pipelines. Amateur knowledge of *Python*, *Java* and *Javascript*: able to write simple software for own needs as well as patching open source software; basic knowledge of *C*, *Ruby*, and *Perl*.
- **DBMS**
 - Intensive interaction with *MySQL RDBMS* in the past, good understanding of database systems maintenance and lifecycle, including replication, backups, and failover. Basic knowledge of *PostgreSQL* and *MongoDB*.

Education

- 2006-2014
 - **Moscow State University of Railway Engineering**
 - *Department of automation, communication and electrification*

Master's degree in information security

Miscellaneous

Besides IT I am also passionate about music making. I play electric guitar, software synthesizers, samplers; I compose tracks, sequences, build layers of effects, create simple visuals, and record demos.

I also like to spend my time reading books, articles, and listening to podcasts about psychology, self-improvement, etc.

Web accounts & social media

- Spotify — <https://open.spotify.com/artist/29YckDRg1YPYeLvRvYBNmN?si=4Rq8INMtRPOGK6IDTDlcAA>
- Instagram — <https://instagram.com/reactorcoremeltdown>
- Website — <https://rcmd.space>