Nikolay Rodionov

```
St. Petersburg, Russia

- phone: +79996690458
- email: nicrodionov@gmail.com
```

About me

I'm an engineer with 3++ years of hands-on experience in different IT areas: from writing e2e auto-tests in JS to setting up a Kubernetes cluster from scratch on bare metal. Just over a year ago, I've completely understood that being a system engineer is what I truly like to do.

A short list of things I love doing

- Setting up, managing, and supporting K8s clusters
- Writing scripts to automate manual actions (Go, Perl, Bash, or Ansible)
- Linux administrating
- Working with containers
- Setting up CI/CD (Gitlab-CI, GitHub Actions)
- Resolving incidents and troubleshooting problems
- "Everything as code (from QA to Infrastructure)"

Expirience

Itigris: ***devops engineer/qa automation***

> *07.2019 - until present I have started as a QA automation and then have moved to the system engineering*

1. As a QA Automation
    - Creating a codebase for E2e tests (**Java**, **NodeJS**) from scratch.
    - Creating and supporting e2e and integration tests **Gitlab C**I pipeline with dynamically starting **Selenoid** server
    - Automating basic actions with **bash**
    - Working a lot with **Docker**, **docker-compose**, and **Dockerfiles**
    - And a bit of SQL quering (**Postgres**)
2. As a System Engineer
    - Writing scripts (**Go** and **Bash**)
    - Supporting and setting up several **k8s** clusters (**AWS EKS**)
    - Supporting services running in **docker** on **ec2** instances
    - Supporting **ec2** instances too
    - Troubleshooting incidents (**k8s**, **nginx**, **aws**)
    - Disaster recovery (**k8s**, **docker**)
    - Maintaing the process of "microservicing" the old monolith
    - Deploying services written in Java, JS and Python
    - And a bit of SQL administrating (**Postgres**)
    - Setting up an infrastructure with **Terraform**

    ### Etersoft: *engineer*

    > *03.2017 - 06.2019 officially (and until present as a side project) I became as a "handyman" and was learning how to do anything in this company.*

- Lots of **Linux**, containers (**docker**, cri-o), and virtualization (VirtualBox, Proxmox)
- A bit of networking (**IPtables** and UFW)
- A bit of **Kubernetes** (setting up and supporting little bare-metal clusters from scratch)
- Setting up virtual machines with **Vagrant** (VirtualBox and a bit **Docker**) and **Ansible**
- Setting up **Nginx** and **Envoy**
- **Bash** and **Perl** scripts for automating basic actions
- Create a codebase for E2e tests (**NodeJs** and the plain **Selenium**) from scratch.
- Create and support e2e tests **Gitlab CI** pipeline with a small static selenium server
- A bit of frontend development (**ReactJS**) so I have a basic understanding of HTML and CSS too
- A bit of SQL (**MySQL**)

Skills

- Kubernetes
    - Kubeadm
    - EKS
    - Docker, Containerd, Cri-O
    - Kubectl, Helm, Kustomize
    - Ingress Nginx, Istio
    - Rook-Ceph
    - Keel, Kube-Monkey
    - Zalando Postgres
- AWS
    - EC2
    - RDS
    - S3

    - Route53
    - EKS
    - Elasticache (redis)
- Coding
    - QA automation (JS, Java)
    - Backend development with Go (GRPC and Rest API)
    - Scripting (Go, Perl, Bash)
    - Frontend development (ReactJS and Elm)
    - Just coding (Haskell, Go, Perl, NodeJS)
- Others
    - Linux, MacOS
    - Docker
    - Ansible, Terraform, Vagrant
    - VirtualBox, Proxmox
    - Nginx, Envoy
    - Prometheus, grafana
    - ELK
    - Postgres, MySQL

Cover Letter

Hi! I'm a young DevOps engineer with 3++ years of hands-on experience in different IT areas. I had experience in several IT areas, and I understand how things may work for different teams, which is important for implementing the DevOps philosophy in the company. After working in a QA engineer position, I understand how to test any solution and describe test cases. Also, I'm not afraid of experiments when it comes to solving unusual problems and using new technologies. I know how things work on bare-metal and in the cloud, so I may help with migrating to the cloud and vice versa. Also, I'm thinking about consequences, and I'm ready to take responsibility for every decision I've made. I've got a will to learn, so I won't be stuck with old-fashioned technologies claiming they are the best fit if there are better alternatives. And I'm never saying that it's impossible to solve any problem unless it is.

When should you pay attention to my CV?

- You working with Kubernetes
- You need to improve/refactor/create/support CI/CD pipelines
- You need help with docker
- You're migrating to the cloud
- You need to support legacy application
- You need help with supporting micro-services application