CV / DevOps
Petr Ospalý

petr@ospalax.cz · Brno, Czech Republic

My name is Petr Ospalý and I am working currently as a DevOps and Cloud Engineer. My career started in the Linux system administration (among other things) and naturally progressed towards the software development – ten plus years of experience in the field. The emerging DevOps movement in the IT industry is bringing the system engineering and the software development together in a way I found intriguing and perspective.

CONTACT

Email:	petr@ospalax.cz
Address:	Brno, Czech Republic
Birth:	1988, Brno, Czech Republic
GitHub:	https://github.com/ospalax
Website:	https://ospalax.cz

EXPERIENCE

SEP 2019 – PRESENT
CLOUD ENGINEER, OPENNEBULA SYSTEMS
Contributions to the open-source project OpenNebula – Virtual Appliances, VNFs, System Tools, Tests, VM Contextualization, Containerization and Deployment of the OpenNebula Docker Image, Documentation and more.

MAR 2020 – PRESENT AS PART-TIME
DEVOPS/SW DEVELOPER, TIETO

SEP 2018 – DEC 2019
DEVOPS/SW DEVELOPER, TIETO
Contributions to the ONAP (Open Networking Automation Platform) open-source project.

OCT 2016 – SEP 2018
SENIOR SYSTEM ENGINEER, SMART COMP
Linux System Administration, DevOps, Scripting, Automation, Docker Containerization.

APR 2011 – SEP 2016
SYSTEM ADMINISTRATOR, EUROSAT CS/EUROSAT HOLDING
Linux and Windows System Administration, Virtualization, Scripting, Automation.

EDUCATION

NOT YET FINISHED – ON GOING (SINCE 2015)

BACHELOR DEGREE, UNIVERZITA PALACKÉHO V OLOMOUCI

Applied Computer Sciences – It's a little challenging to study CS while working full time jobs.

2009

UPPER SECONDARY, GYMNÁZIUM (GRAMMER SCHOOL), BRNO, VÍDEŇSKÁ

Specialized programming focused class.

LANGUAGES

CZECH: NATIVE SPEAKER

ENGLISH: FLUENT

SKILLS

Unix, Linux, Czech, English, Cloud, OpenNebula, Docker, Kubernetes, Python, Ruby, Shell, Bash, Sed, Awk, Scripting, C, Backup, PostgreSQL, Git, Gitlab, System administration, Automation, Virtualization, KVM, Nginx, Docker-compose, Monitoring, Nagios, SaltStack, Ansible, Powershell

- **LINUX SYSTEM ADMINISTRATION**
  - Debian/Ubuntu/RHEL/CentOS/Alpine
  - Generic knowledge of the GNU/Linux OSes and Unix-like environments
- **SHELL SCRIPTING AND AUTOMATION**
  - Development of shell scripts and system utilities (in bash/dash/ash, grep, sed, awk etc.)
  - Automation scripts (Let's Encrypt renewals, cron jobs etc.)
  - Ansible and SaltStack Automation (Custom python modules for Ansible)
  - Powershell scripting for Windows
- **PROGRAMMING AND DEVELOPMENT**
  - C programming
  - Python, Ruby
  - Git, GitHub
- **CLOUD, VIRTUALIZATION, CLUSTERING AND HA**
  - OpenNebula, KVM, Qemu, Libvirt
  - Clusters (PostgreSQL, Etcd, Keepalived etc.)
- **CONTAINERS, MICROSERVICES AND ORCHESTRATION**
  - Development of Dockerfiles, Entrypoint scripts, Docker-Compose deployments, Kubernetes manifests etc.
  - Docker and Docker Swarm
  - Kubernetes
- **CI/CD**
  - GitLab (Travis CI)
- **DEPLOYMENT OF OTHER SERVICES AND PLATFORMS**
  - Web Servers (e.g.: Apache, Nginx)
  - Databases (PostgreSQL, MySQL/MariaDB)
  - Information Systems, CMS, WebApps etc.

ABOUT ME

I like the world of Linux and open-source in general – the free (as in freedom) options and the possibility to fix or create stuff by yourself for yourself. Developing scripts or programs is both fun and joy. I have always an urge to automate stuff and make the system self-sufficient – by creating robust tools, utilizing HA, clustering and self-healing mechanisms. Writing the procedures down in a smart way removes the need to memorize them or research them yet again... Automation thus frees my hands to do more and my mind to cramp there more important knowledge.

IT is a demanding and challenging field and it will require never ending learning and studying cycles – I love to learn new skills and gain new knowledge. Building or creating something (code or system) is rewarding and pleasant experience for me.

My over a decade long experience stems from the System Engineering background which always required a lot of coding to achieve the desired goals. Therefore I have the advantage to know which approaches are ineffective or conceptually wrong. The system is not a black-box to me with an infinite compute power or memory.

The recent popular DevOps movement is not just a fashion trend which will go away as fast as it arrived. I truly believe that any developer should know ins and outs of not only own developed piece of software but also the whole environment in which it will be running. Developer then will quickly know and understand what works well and what is a bad design or an error - just by owning the complete deployment.

If I have to mention some things what I don't like then it would have to be any unnecessary cognitive load like dealing with yet another niche language or with buggy reinvented wheels or other forms of Not-Invented-Here syndroms. It is also disheartening to work on a clearly mismanaged or badly designed project which will be either shelved eventually or it will become a theater of a never-ending bug hunting.