Basic information

Basic information

Name	Kerola, Sami Matias
Date of birth	April 28th, 1977
Place of birth	Raahe, Finland
Marital Status	Married
Driver's Licence	Yes, category BE
Military Rank	Lance Corporal (korpraali)

Contact information

Address	Flat 4, Clarion House
	70 Park Hill
	London
	SW4 9PB
	United Kingdom
Mobile	+447795344427
Email	kerolasa@iki.fi

Language skills

Finnish	native
English	fluent enough
Swedish	bare minimum
Dutch	few words

Work experience

Cloudflare – July 2015 onwards.

Couple months after lastminute.com was bought things started to look like company acquisition related lay-offs are coming, so I moved on. *London* at this point still felt like a reasonable place to live, so I decided to find something else from the same town. While looking Cloudflare got my attention as a company that is using *golang*, so I decided to send a CV to them and see if I could get a programmer position. That did not turn out to happen, but I got *systems reliability engineer (SRE)* position.

When I joined Cloudflare it had ridiculously large installation, it handled more requests and traffic I've ever seen. During my time in Cloudflare the company has doubled everything. At September 2017 scale is over 100 installation locations, thousands of servers, 2 terabits of traffic flows, and 2 million requests per second. As an SRE my duties has included *on-call shifts*, *attack mitigation*, *salt stack* configuration management, and *release engineer* work. Due the sheer number of releases I have taken care of I was asked to formalise *release process*.

There are lots of smart people in Cloudflare, who are doing all sorts of clever things. I cannot say I am familiar with all what is going on, but during my period here I did come familar with *Grafana*, and *Prometheus* monitoring, *Perceus provisioning*, and *Circus* process manager. I wish I would have had more time to get familiar with other technologies in use, such as Kafka, Mesosphere and Marathon, Apache Flink, and golang, but my job was only to see from users perspective how these work.

One of the takings from Cloudflare is an experience to be part of company that is structured to follow the sun setup. The company has SREs in London, San Francisco, and Singapore. That means three handovers per day. Lesson never gets old, it is the clear communications that make and break days. It really does matter what to be verbose and accurate in *git* commit messages, *HipChat*, *Jira tickets*, and in *Confluence* documentation. I am pretty sure my influence to Cloudflare has been positive with making others to follow my practises, such as not assuming people can read your mind.

lastminute.com – September 2012 to June 2015.

The Netherlands started to feel like a place I've seen. For whole of my life I've been moving from smaller places to bigger, and found them more interesting than the previous. With that in mind it really should not be great surprise I went to *London*.

When I search a job from London I was open to any reasonably sounding company, and lastminute.com found me. In the lastminute.com I have duties from typical *senior system administrator* tasks to an

architect type *role*.

At the time when I joined *puppet* management was still new for the company. Now the existing systems has been either reinstalled or retro fitted to use puppet, *foreman*, *pulp*, and some systems even do continuous deployments. Needless to say cleaning up a decade of legacy is a big on going job, with a twist that once something starts to work one finds more broken items to fix.

The introduction and proper use of configuration management, and advanced facilities for developers such as *jenkins pipelines*, has made it clear there is lots of room for improvements. The areas where improvements are mostly needed are *monitoring*, *inventory maintenance*, *account management*, and *orchestration*.

Due my experience in open source my managers has asked me to drive many of the improvement efforts in *systems designer role*. That has lead my work time to be split in three more or less equal effort categories; *keep* the other system admins and developers in *align* with *long term goals*, try and document *new stuff*, take a part of *problem solving* when old system has *big impact issues*. Basically I am halftime tech coach part time developer researcher fireman.

The lastminute.com job has made me to learn a lot of new stuff. The existing puppet skills has been updated to whole new level. I had to learn some *ruby*, and relearn *subversion* in ninja level. This is also the first time when I am so aware of *PCI compliance*.

One of the new aspects in this job for me is dealing members of the team in India and Texas. Having *team so far apart* in timezone wise is not easy, but there is no other option than deal the situation and be as *clear* in *communications* as possible.

My final deliverable at lastminute.com end up being a technical plan to migrate data centre migration from US to Europe, with emphasis on infra structure services, such as provisioning and configuration management. Luckily by the time planning took place the puppet master was already using *gitlab* and *r10k* so that managing complex view of data was at leas possible if not easy.

TomTom – September 2008 to August 2012.

I always knew I wanted to go abroad. It took me nearly a year to find a position that would work for both me and employer, and all matched with TomTom.

When working in TomTom my primary job is to maintain ⟨http://routes.tomtom.com/⟩ site. The job included *change management* from test to production, *developer support*, being *third line customer support*, setting up *monitoring* and looking after *security* and *performance metrics*. This role included a lot of interaction with different developer teams, who made the web map and traffic information systems. Without knowing I was doing *devops*.

But being part of devops did not mean I could stop support activities in an *on-call rota*, or being part of various small Techops organisation *projects*. Such projects included two *data center moves*, taking *puppet* into use, reconstruction all *documentation*, and organisational stream lining that happen in form of tearing down lots of red tape. I also did reimplementation of all *DNS* management and server setup.

In TomTom almost all systems are *Linuxes*, network equipment *Cisco* and storage *NetApp*. In TomTom nearly all applications are run in java containers, which are 60% *jboss*, 30% *tomcat* and rest are various such as *daemons* wrote in *C*. In this company I got to be very familiar with *Confluence* and *Jira*.

Capgemini Finland – January 2007 to August 2008.

After the telco experience I went to *Capgemini* to checkout how it feels to be a consultant.

I was tasked to be *principal engineer* in implementing retail company *order process systems*. At that time the system handled orders worth of *billion euros per year*. When the retail system was done my second task in Capgemini was an internal *DNS architecture* for *Nokia Siemens Networks* . The goal to make ridiculously highly reliable system, so I told them how to make *anycast* DNS to work.

While being tasked with fairly interesting client jobs I also took part in *on-call rota*, and participated in normal system administration activities for various big and small customers. That familiarised me with *AIX* on *IBM z-series*, *HP-UX* of all sorts using hardware from blades to *superdome*. I also got familiar with *Sun Cluster* , *HPUX Serviceguard* and *HACMP* on AIX. The *Oracle RAC* setup in HP-UX environment using disks from *HP EVA* with *multipathing*, that where bound together with *volume groups* and *logical volumes*

made me first pull my hair off, but now when I think the stuff the experience really thought me how to deal *block devices*. Perhaps I should be happy that I needed to take care of that mess as well.

TeliaSonera – April 2000 to December 2006.

Sales support was not really my thing, so I started to look for more *proper system administration* jobs. The company who took a change with me was *TeliaSonera*. This is the point in time when I consider my professional system administration career to begin. The first three or four years I was a *web portal administrator*. In the early days portal was made of *Vignette V4*, that I upgraded to *V5*. Loadbalancing was done with *Alteon switches*, that were there before my time. Later on the portal got *Bluecoat* reverse proxies were used for http acceleration, user authentication was moved from content database to *Sun Directory LDAP*, along with various other enhancements that most people consider now a days as standard web building blocks. For me it was great to be part of team that was among first ones in the Finland to do this sort of stuff.

When I moved from Tampere to Helsinki I got a change to become a *DNS hostmaster team* technical lead. My duty was to *maintain*, *upgrade* and finally *redesign a DNS platform* containing 20.000 domains and about 1 gigabytes of recursion traffic per hour. I also took care of answering to hostmaster requests that required in depth *RFC* knowledge. These DNS systems ran on *Linux,and*had *F5 BigIP* load balancing.

RTT Ohjelmistopankki – June 1999 to March 2000.

The RTT Ohjelmistopankki was my first real job. In this company I helped the *sales* as a technical person. Company was selling mostly computing hardware to small business near by, and education facilicies all over Finland. Later category was the reason why technical people were needed.

Novo Group – February 1999 to June 1999.

In *Novo Group* as system admin trainee I encountered *OpenVMS* which I really could not quite understand. Especially the *DECNet* was something that I struggled, but apparently I made good impression to my boss as I got some responsibilities to do *Digital unix* work. I also did some *microsoft windows* support, and this has been the last time I've done that during work hours.

Open source activities

dhcpd-pools

The *dhcpd-pools* was my first real open source effort. In *TeliaSonera* we had a problem of having rather large dhcpd setup. Monitoring the state of leases was about taking nearly as long as polling interval. I decided to solve the problem. First implementation was *awk* script that proved the algorithm I chose worked, and the second version was wrote in *C*. When the analysis program ran as expected I decided to publish the work.

I have maintained this tool since 2006. The dhcpd-pools has been really great project for me, and lead to know how to use *automake* and *autoconf*, static analyzers such as *gcc* with all warnings, *clang*, *smatch*, and *cppcheck*. The numerous times I've caused a crash has teach me usefulness of *gdb*, and *valgrind*. For performance tuning I first used gcc profiling, before learning how to use *perf* tool. And obviously version control systems has been in use since early days, first *cvs* then *git*.

util-linux

My very first commit to project.

commit c49e31f4ee7c35946f91335a8a3179f503bd9cf0
Author: Sami Kerola <kerolasa@iki.fi>
Date: Fri Oct 1 00:33:44 2010 +0200

That makes beginning of real open source activity. Being member of this project has changed a lot how I think about computers, programming, and co-operating with community. I have even caused a CVE-2015-5224, and fixed one CVE-2015-5218.

progps-ng

It was ages ago when I sent a patch to procps to add −−human readable option to *free*(1) command. The upstream was dead, and nothing happen. Then I noticed RedHat, Suse, and Debian had forked procps, so I sent the patch to them and it was accepted in April 2011.

By summer 2012 I had done about two hundred changes to project, that fixed *build system*, unified *coding*

style, added *long options* to commands, made *manual pages* clean, enhanced *error printing*, made *NLS* to work and so on. At this point the libproc was known to be next thing requiring a lot of changed, but the project seemed to lack an idea how to do that. Meanwhile I end up doing some other contributions, and the procps-ng eventually was out of my radar.

f5gs

The *f5gs* has similar history as *dhcpd-pools*. A company I am working in, *lastminute.com*, had a problem, and I chose to fix the issue with a piece of software. The technical problem again is pretty simple. It would be nice to control whether a load balancer, *F5 BigIP*, is sending traffic to a destination with a tool that is in destination. Or to say the same in different way; how can I *declare* a server to be in *maintenance state* without logging in to various web interfaces here and there.

The f5gs is a *daemon* that answers to *socket* telling a *state* that was declared by human being. The socket can answer finite number of states, which means they can be expected, and one can make F5 to use this daemon as *health check*. Tool also allows run of state change scripts, that will perform tasks *managed with puppet* and relevant for the system. An example of such is suppression of *alarming* when maintenance is declared.

GNU Hello

After sending few patches to *GNU Hello* the maintainer asked would I like to the project. I was ok with that, so now I am a official *GNU project maintainer*.

Education

Further Qualification in Information Technology

Sep 15 2004 - Feb 09 2006 Amiedu

Those who have completed the qualification may work in different assignments within IT companies or as people responsible for the use of PCs and LANs in public administration, companies and other corporations.

Raahe Commercial College

Aug 14 1996 - May 31 1998 Raahen Porvari- ja Kauppakoulu

Senior secondary school

Jun 01 1996 Saloisten lukio

See also

Links

My homepage ⟨http://www.iki.fi/kerolasa/⟩.
Linkedin profile ⟨http://www.linkedin.com/in/kerolasa⟩.
Open Hub profile ⟨https://www.openhub.net/accounts/kerolasa⟩.
This resume in git format ⟨https://github.com/kerolasa/cv⟩.