



Kelly Becker-Neuding

Generalist Full-Stack Software Engineer
Development Oriented Operations Engineer
Service Oriented Software/Operations Architect

Address	45 Mayfield Ave. Daly City, CA 94015
Email	kbecker@kellybecker.me
Cell Phone	(415) 871 - 3657
Website	http://kellybecker.me
Blog	http://objectiveoriented.com
GitHub	http://github.com/KellyLSB
LinkedIn	http://rain.kellybecker.me/6l4w
Launchpad	http://launchpad.net/~kellylsb

Professional Summary / Cover

I am a hyperactive software genius!

Over the past 6 years in the profession, I have had the chance to work on everything from wiring cable at layer 1 all the way up to building applications at layer 7. While I do claim the title of "genius" I do not use the default interpretation. In my life experience I have noticed many believe genius' are smart while in fact it is about intelligence. I cannot claim to know everything but I will claim that I can learn anything at a highly accelerated rate.

Growing up with untreated ADHD caused me to develop some coping mechanisms; software was a haven from the overwhelming world around me. This flaw of mine in the everyday world has since turned into a giant merit in the software industry. I always strive to architect, build, provision, code and deploy in the most optimal way within the bounds of my knowledge. And physics, of course. Though if I can do better and there is time I always try to build the best and eliminate technical debt on every layer.

Since beginning treatment my ability has skyrocketed; I try to read a new technical article on a daily basis as well as architect or work on a new idea. I also have become a lot more aware of myself and I have been working to turn each flaw into a new merit. The road has been tough and I look to my career and the systems I work with to help me learn organically, through my day-to-day life. I build meta concepts about communication and the people I interact with to help me be a better contributor to society whilst understanding the world better.

I live to code. If my skills and my person sound intriguing then I would love to talk with you.

Quick Contents

Professional Summary / Cover.....1

Philosophy.....2

Work Experience.....3

Languages and Technologies.....?

Understanding The Skill and
Technology Clouds.....?

My long term 10 year goal is to accrue enough experience and connections to start and run my own company.

Notable projects I am proud of:

- A custom Linux Kernel built for the Acer Aspire One AO521 to support the ACPI peripherals properly (2010)
- The EvolutionSDK Modular PHP Framework along with my good friends and colleagues [Nathanael Ferrero](http://nateferrero.com/) (<http://nateferrero.com/>) and [David Boskovic](http://www.david.gs) (<http://www.david.gs>) *at time of writing David's site was unroutable* (2011)
- Full PCI Compliance passing a [Trustwave](https://www.trustwave.com) (<https://www.trustwave.com>) Intusion Scan for financial information with Red Hat Enterprise Linux 5 on a Rack Space Hosted Private Network with a DMZ (2011)
- The Rails "Gemification" process which takes large components of monolithic Rails apps and turns them into mountable plugins that can run as part of an existing application or standalone with communication through dRB or a Restful API (2013 - Continuous)
- Docker.IO Service Oritented Architecture blue prints with Dynamic DNS for service discovery (2013 - Continuous).

Philosophy

No one is perfect and everyone has baggage; this is something no one in our world can deny. I believe that honesty is the best policy. I'm a firm believer in our freedoms, especially the freedom of speech; however, with all freedoms they must be exercised in moderation. Because of this and my placement of importance on honesty, I will always be upfront and direct because there is a reason for everything and we are all human we make mistakes and sometimes do not think things through. This happens both in the scope of personal issues and the importance we place on our careers. I know that was a long lead in for my primary philosophy.

... Because this life is yours. Some of it was given to you, the rest you make yourself.

This statement paints a beautiful picture of how I view life and my career. I'm not perfect both due to situational circumstances of my past and present but I have the power to change and be different.

Work Experience

Jiff, Inc2013 - 2014

TitleSr. Software Engineer
LocationPalo Alto, CA
DurationSeptember 2013 - April 2014

Details

At Jiff I was responsible for the Rails Platform. I was part of a team of 4 total engineers; made of up of our Lead Engineer, another Sr. Engineer, a Jr. Engineer and Me.

We were constantly working on making our Ruby backend to be more responsive and more optimized. Our long term goal was to reorganize our platform code to be more fit for a Service Oriented Architecture. We started by optimizing our pain points that used heavy queries, duplicated code and overly complex data models. Then we spent time determining which components were specific to services we offered; then the hacking began and I mean literal hacking we cut out close to 75% of the code base and we still had more to go when I left.

One of the solutions we were using to migrate to SOA so that our services could run independently of the platform was called "gemification". While the idea was the team's, I was given the honor to come up with the architecture and implementation process; which was subjected to a through peer review by the Platform, API and QA teams. This process included a write up, prototype and benchmark. I then helped guide the each team member in ensuring that we were consistent with our decided plan and did not deviate without a proper meeting; a common issue at Jiff.

Along side this plan, I proposed a PaaS server architecture using Amazon VPC, EC2 and Docker.IO with Bind9 Dynamic DNS for service discovery. There was much team backing and there was even time that was allocated to the implementation process as well as a

Technologies

Ruby / Ruby on Rails

Proprietary Gems

Open Sourced Gems

Restful JSON

Sidekiq

Rspec

Capistrano

Bash / SH

Puppet

Solr

GitHub

Circle CI

Code Climate

Airbrake.io

Graylog 2

Amazon AWS

Amazon AWS - VPC

Amazon AWS - EC2

Amazon AWS - ELB

Amazon AWS - SES

Amazon AWS - AMI

Amazon AWS - EBS

Amazon AWS - S3

Ubuntu 12.04 LTS

Docker.io

Bind9

IRCD2 (Admin)

Stripe

Redis

MySQL

More...

working prototype. Unfortunately our new VP of Engineering had cancelled the project before it's completion.

Along side gemification I was a huge advocate for taking non-company critical code and releasing it open source. We released 2 gems during my duration at Jiff, unfortunately there was not enough time to properly document them to my satisfaction. Upon being let go I forked them and plan to continue development on them.

Projects

- Gemification - Details Soon
- SOA Migration - Details Soon
- Docker.IO PaaS Infrastructure - Details Soon
- Polytag (<http://github.com/KellySB/Polytag>) - Details Soon
- Multitype (<http://github.com/KellySB/Multitype>) - Details Soon
- Dynamic CSV Validator - Details Soon

Youmacon

2009 - 2013

A.K.A.	Defying Conventions
Location	Detroit, MI Remote
Titles	Registration Software Architect Website Server Administrator Director of Pre-Registration Assistant to the Chairman
Duration	June 2009 - November 2013

Details

Youmacon was my first real big job. I had been doing some small jobs writing custom Content and Customer Relation Management Systems for years. I was approached to architect, build, deploy and manage a ticket sales

Technologies			
EvolutionSDK	PHP	PHP-FPM	
MySQL	Stripe	Postmarkapp	
Amazon AWS RDS	Amazon AWS SES		
Amazon AWS EC2	Amazon AWS ELB		
Rackspace Public Cloud			
Dreamhost Shared Tenancy	Amazon AWS AMI		
Amazon AWS S3	Ubuntu 10.04 LTS	PhpBB	
HTML	CSS	Apache	Apache Mod PHP
Centos	Nginx	PayPal API - SOAP	
LHTML	Graylog 2	Joomla	HAproxy

application for a ~5,000 person convention. I jumped on the opportunity and proved myself very quickly.

The initial system was really nothing special. It was built from scraps of my old projects and had a really bad design, but it was functional. It integrated with the PayPal SOAP Api to process the ticket sales while providing a backend that could be used to handle refunds, complimentary tickets and discounts.

Over the years it started becoming a platform and I modularized all the components, bought the rights, contracted it back to them and named it ¥ENN rEvolution. The new platform had so much more; it had gifting, stripe integration, convention staff management, and newsletter aggregation to get our attendees all excited for the upcoming year (using Amazon SES). It was and still slightly is one of my biggest prides.

The platform was built on EvolutionSDK V3; the same framework built by the teams at Verschoye and Momentum OS, I felt since it was a framework I thoroughly understood it would be the best for the job, and it was amazing running on top of PHP-FPM and Nginx.

I resigned in December 2013 as the convention grew so large I was no longer able to manage the pre-registration and since the convention has primarily volunteer staff I was not able to consistently get help. Though depending, since I know they need it and the Chairman won't stop asking me, I may go to the convention and run his registration department on site like I had been. He usually needs someone dedicated and loyal to his business needs to help put out the fires.

Recomendation

Since Kelly's introduction to my staff, my event has advanced far beyond its previous scope through the systems she developed for us. With these systems, we can process information and registrations with far greater efficiency and efficacy. Thanks to her efforts, our capabilities are far greater than those of our competitors, and our event has greatly expanded as a whole.

— Morgan Kollin, Chairman - Defying Conventions (<https://www.linkedin.com/pub/kelly-becker/20/830/67b#recommendations>)

Rainforest QA

2013

A.K.A. Cldrdr, Inc

Location SoMa, San Francisco, CA

Title Inapplicable

Duration August 2013 - September 2013

Details

Rainforest QA was an amazing opportunity.

My only regret is that it was so short lived. It was a small 4 person company in SoMA. We were working to better user testing and browser testing and eventually automate it.

While I was there I spent pretty much all my time cleaning up technical debt. My first week was going through every Rspec test and ensuring they passed and switching from fixtures to factories; this was a challenge as this was actually my first proper introduction to unit tests. The second week was doing the same on front end jasmine tests, including writing a factory library for Node.JS and Javascript, called [Industry](http://github.com/KellyLSB/Industry) (<http://github.com/KellyLSB/Industry>): factories targeted for use with Backbone.JS.

Later I worked with Russel, the CTO, to architect a cascading virtual machine cluster with Virtual Box and Vagrant to create IE-VMS that we could use to ensure the accuracy and authenticity of the tests returned by our Amazon Mechanical Turk users. This process involved testing the installation on close to 10 cloud providers. Long story short, it was not worth trying to do on an already virtualized machine and in the end, though it was more expensive, we used on-demand Softlayer bare metal.

In order to monitor and recycle containers I wrote a Grape API on top of Sinatra to be a communication handle with our Platform as well as implemented webhook functionality to poke our platform as the machines changed states or events were fired. In order to provide monitoring of the user's actions I wrote a plugin called vagrant-camera which took a screenshot every time a user made an interaction with the virtual machine. The plugin would then fire a webhook back to our platform sending the image as a base64 jpeg.

Technologies

Ruby Ruby on Rails Heroku Circle CI

Javascript Node.JS Backbone.JS

Grape API Sinatra Vagrant

Virtual Box Digital Ocean Azure

Softlayer Jasmine Rspec Amazon S3

Amazon Mechanical Turk Postgres

Triggit, Inc

2013

Location Mission, San Francisco, CA

Title General Support / DevOps Engineer

Duration January 2013 - July 2013

Details

Triggit was my first DIY shop. They were all about not telling you what to do and just having you jump in the deep end, often without teaching you to swim. I actually realized an issue that became really apparent after the next technical hire after me; they never actually gave me "training" or even documents showing their stack. I actually had to explore and figure it all out myself. Considering this and the complexity of the Triggit stack I think I did a really good job of that.

I was hired to be a technical liason between the customer support team and the rails self-service platform. After hiring me they actually seemed confused on what I would do, so I did a couple Ruby tickets, then got bored. After asking what I was supposed to be doing a couple times I got put to work optimizing and updating Hive queries for the reporting exports as well as running manual reports or even sraping the documents directly from Hadoop.

Evtually I needed a break from the menial tasks. I went to Ryan (CTO) and asked him if there was something else I could work on to un-burn myself out. He said he could not think of anything, but if I found anything or thought of anything that I should let him know. The next day in the morning roundtable Ben, our operations guy, mentioned he was extrmely backlogged. I offered to help; this was my official introduction to Dev Ops.

He showed me the tools and scripts he was using and I could not make head or tails of them; they were all aliased with short acronyms. I think his puppet run scripts were `upr` or something like that. Many were shorter and ran operations on multiple machines and were potentially dangerous with no organization or documentation. I quickly understood why he was backlogged. The first thing that I noticed was our puppet manifests; I had some small experience with puppet before, but nothing major and even I knew that this

Technologies

Ruby Ruby on Rails **IRC** Clojure C
Hadoop **Hive** Cassandra **Postgres**
MySQL **Redis** **Amazon S3**
Datapipe Metal Datapipe Cloud
Bash / SH **Genders DB** **Puppet**
Ruby / Net-SSH **GPG** Nginx Apache
Go-Lang **HAproxy** HTML CSS **Javascript**
Coffeescript

was horrendous. There were 250 server declarations and every single type in use were all declared in the same file. To add insult to injury the entire file had mixed indentions and appeared to be minified. There was a lot of work to do.

I sat down with Ben and determined that he was a classic system administrator and he started having trouble managing the server stack after about 50 servers and was to prideful to ask for help. I don't say this to say he was an arrogant man, most of the time, but rather to emphasize the challenge that was at hand. My first week helping him I had a goal; properly indent and organize his puppet manifests, changing nothing. I ran it by him and Ryan and very quickly was given the approval, so I got to work. The process took about two days with me working on it from about 10 - 8 every day. After eventually merging it in piece by piece (due to heavy peer review; though nothing changed, puppet noop showed that) we were ready to roll.

The next steps were to remove the duplicate puppet classes and fix the dependency ordering as well as start using community puppet modules in place of some of our not-as-well written or documented custom classes. This process took about two months and then little bits here and there, and that process was still going on after I left.

We also had to tackle how Puppet was running; Ben insisted on using a master-slave configuration that ran only on demand. To work with this stipulation and the delays it caused when trying to update all 250 servers at once I had to get creative. I wrote an early version of what is now my [cli_tool](http://github.com/KellyLSB/cli_tool) (http://github.com/KellyLSB/cli_tool) project, though the original at Triggitt had much more features. The app was in ruby and handled booting up the puppet masters in the given data centers for the server tags we were running puppet on. Once the masters were started it would then boot up each slave server individually, run a noop, dump the changes to a markdown file and stdout, then ask if it should continue. It had a force yes option on the CLI as well, but it was rarely used due to stability issues. This tool allowed up to move from undocumented aliases to an application that ran on the developer's machine instead of server side and had an api like so `puppet-tool run dc newyork` which would run all the puppet clients in the newyork data center. The tool even would dump a log of each transaction into the current working directory where the log was run, broken up by the server hostname and the date. The goal was full transparency.

Along with all this I wrote a backup daemon. This daemon would listen for incoming connections from a FIFO on it's host, then would run the selected library and name, backing up whatever the library was meant to backup. The whole application had a standardized API and I wrote libraries for MySQL, Postgres, Redis, Cassandra and our

configuration files. The server would then process the data as requested by the library and then the library was expected to return a single file. This file would be bziped with libbzip2 (if not already compressed) and then would gpg encrypt the file and upload it to s3 or output it to a specified directory. The daemon also had hooks for the CLI tool I wrote that would download and decrypt the latest (or latest to a date provided) backup of any of the backup types and either A restore it or B download it to your dev machine and optionally import it locally.

While I was employed we had a lot of issues with the network in the office, mostly due to the router dying and all the engineers being on wireless. There were also concerns that we should not have our guests on the network, because of direct VPN connections to our data centers. In order to help with this issue I set up a new network by running new cables to the engineers as well as building and installing a new router capable of easily constraining machines to specific subnets as well as traffic controlling. To build this network I got us a proper UPS to ensure that all our critical communications hardware would be sufficient as well as a [Soekris net5501](http://soekris.com/products/net5501.html) (<http://soekris.com/products/net5501.html>) which I installed [pfSense](https://www.pfsense.org) (<https://www.pfsense.org>) on. This new router was setup to support roaming wifi as well as adding a guest network that was separate from the rest of the network, ensuring security of our servers, local hardware and development machines.

I also was fairly active with company extracurricular events and was a party member of the wednesday night Dungeons and Dragons group (I was the tank). I put all my points into Dex and Strength with near zero intelligence. The party always sent me in first to absorb damage, but my character was known to scream "it's dark in here, I can't see anything!" This, of course, alerted all the monsters in the dungeon. Haha, good times!

Recomendation

Give Kelly a goal and she'll plot the straightest course to it. She can code up an solution and do what it takes to get buy-in from the team. And she's great at picking up new technologies and teaching herself what she needs to get the job done!

— Ryan Tecco, CTO - TriggIt, Inc (<https://www.linkedin.com/pub/kelly-becker/20/830/67b#recommendations>)

Tioki

2012 - 2013

A.K.A.	Demo Lesson, Inc
Location	Lake Merrit, Oakland, CA
Title	Full Stack Software Engineer
Duration	July 2012 - January 2013

Momentum OS, Inc

2010 - 2012

Location	Remote (2010 - 2012) Los Angeles, CA (Nov, 2011) Colorado Springs, CO (2012)
Title	Full Stack Software Engineer
Duration	June 2010 - July 2012

Verschoye Innovation Corp

2010 - 2011

Location	Remote
Title	PHP Contractor
Duration	January 2010 - June 2011

A.K.A.	Defying Conventions
Location	Novi, MI Remote
Titles	Registration Software Architect Website Server Administrator Director of Pre-Registration Assistant to the Chairman
Duration	March 2009 - May 2011

Location	Grand Rapids, MI Remote
Title	Webmaster
Duration	June 2010 - June 2011

Language and Technology Clouds

Languages

bash/sh

puppet

dockerfile

sql

ruby

php

html

node.js

javascript

coffeescript

less-css

sass

css

html

python

elixir

go-lang

clojure

c

assembly

Operating Systems

mac-os-x

ubuntu

debian

fedora

centos

redhat-enterprise-linux

windows

Caches

redis memcache

Frameworks and Libraries

ruby/rails ruby/padrino
ruby/sinatra ruby/active-record
ruby/active-support ruby/net-ssh
ruby/net-sftp ruby/thread
node/express
html-css/zurb-foundation
html-css/bootstrap elixir/ecto
php/evolutionsdk
php/codeignitor php/wordpress
php/joomla

SQL Databases

mysql postgres sqlite aws-rds
hive

NoSQL Databases

mongodb couchbase hadoop

Other Databases

cassandra

How To Read My Skill / Technology Clouds

I decided to use skill clouds to show my expertise and use of a specific technology. I felt that this would help the things I love and have worked with frequently "pop-out" when looking over my resume at a glance. There are 5 categories though depending on the section you are viewing they should have different interpretations. Don't worry they are almost the same.

Work Experience Technology Cloud Map

The technology clouds under work experience show the technologies I used often.

1. Red, Very Small Font: Technology was present; no direct interaction.
2. Grey, Small Font: Technology was present; extremely little interaction.
3. Black, Normal Sized Font: Technology was present; little interaction.
4. Blue, Normal Sized Font: Technology was present; average interaction.
5. Violet, Large Bold Font: Technology was present; major interaction / ownership.

Skill Cloud Map

The technology clouds under work experience show the technologies I used often.

1. Red, Very Small Font: Known; no interest.
2. Grey, Small Font: Amatuer; less then 25% expertise.
3. Black, Normal Sized Font: Professional; greater than 25% expertise and less then 75% expertise.
4. Blue, Normal Sized Font: Highly Professional; Greater than 75% expertise.
5. Violet, Large Bold Font: Almost Expert; Around ~99.9909% expertise.