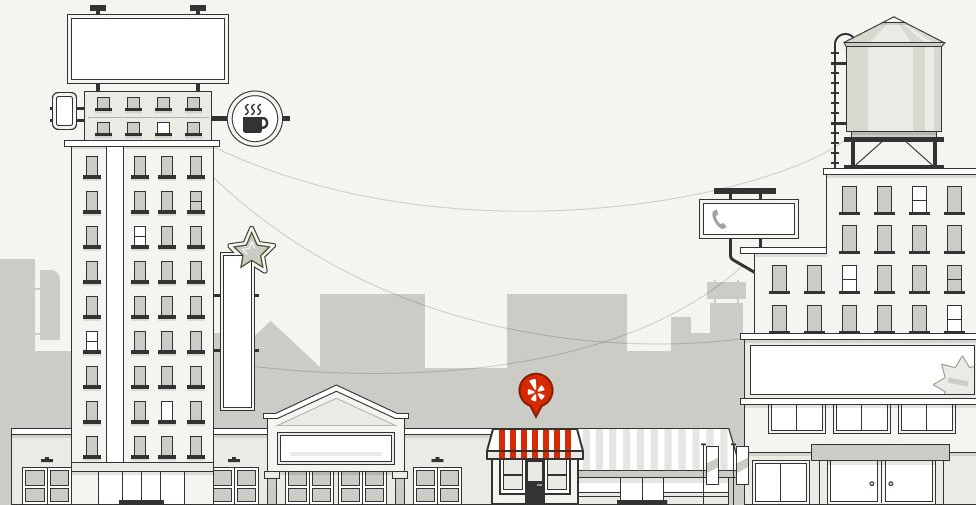


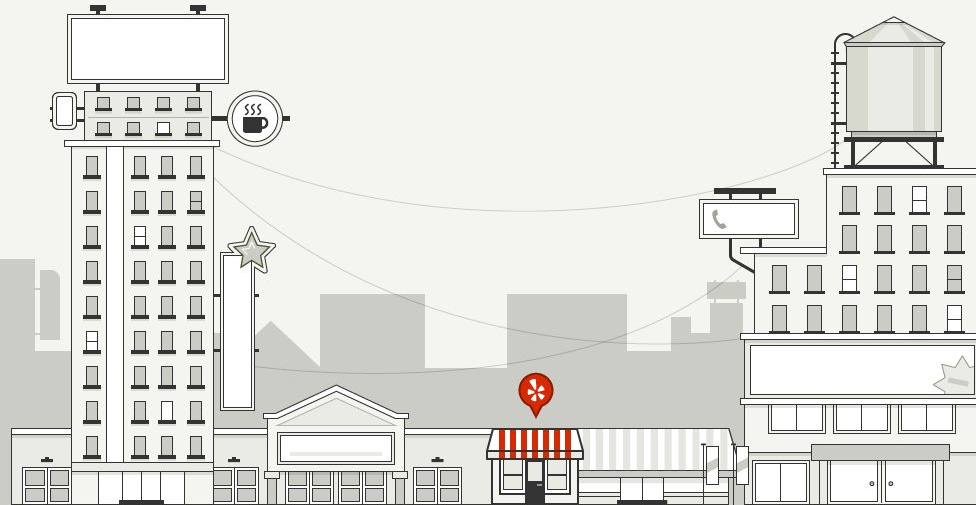
# How to Succeed in Ops Without Really Trying

Chastity Blackwell  
[cblkwell@yelp.com](mailto:cblkwell@yelp.com) / @Black\_Isis



# Flailing Wildly at Feelings of Inadequacy

Chastity Blackwell  
[cblkwell@yelp.com](mailto:cblkwell@yelp.com) / @Black\_Isis



# Yelp's Mission

Connecting people with great local businesses.



Who am I? Why am I here?



# Myths and Realities



# The Myth

“I’m just a system administrator -- I’m not a ‘site reliability engineer.’ I don’t even know what DevOps really is.”



# “NoOps Engineer”

- Configuration management
- Incident response
- Troubleshooting at multiple levels of infrastructure (OS, network, application, database, etc)
- Shell scripting/programming
- Monitoring and metrics systems
- Performance and capacity management



# The Reality

Most of these new job titles are just a new name for something good sysadmins have been doing for years. DevOps isn't about your job title.





# The Myth

“Everyone is looking for a coder. I don’t code.”



```
#!/bin/bash
```

```
REGION='us-east-1'
```

```
HOSTFILE='/home/cblkwell/hosts.txt'
```

```
if [ -z "$AWS_ACCESS_KEY_ID" ]; then  
    echo `AWS_ACCESS_KEY_ID not set`  
    exit 1  
fi
```

```
for host in `cat $HOSTFILE`; do  
    aws ec2 describe-instance --region=${REGION} \  
    --filters "Name=tag:hostname,Values=${host}"
```

```
class webapp::config {  
  
  file { ['/service/webapp/config':  
    ensure => file,  
    owner  => 'webapp',  
    group  => 'webapp',  
    mode   => 0644,  
    content => template('webapp/config.erb'),  
    Notify => Class['Webapp::Service']  
  }  
  
  file { ['/service/webapp/logconf':  
    ensure => file,  
    owner  => 'webapp',
```

```
#!/usr/bin/env python
```

```
import boto
```

```
conn = boto.ec2.connection.EC2Connection(aws_key,  
aws_secret, region=region)
```

```
def find_instance(ip_address, conn):  
    instance = conn.get_only_instances(  
        filters={"private-ip-address" : ip_address })  
  
    return instance
```

```
def process_file(filename, conn):
```

# The Reality

You already write code, you probably just don't think about it as "real programming."



# The Myth

“I run some webservers and a MySQL database for a small company. I can't compete with someone working for Google or Facebook.”



# The Reality

- Not everyone is Google or Facebook
  - 6.7 million tech employees in the US in 2015
  - Google only had ~62,000 employees worldwide
  - Facebook had ~12,500 employees worldwide



# The Reality

There are a lot of advantages of working at a small shop.

- Less likely to get stuck in a bubble
- More exposure to different parts of the application stack
- Can make a greater impact and get better feedback





# The Myth

“I can't apply for a job at that company. Jane Shepard works there, and she literally wrote the book on operations. Why would they hire me?”



# The Reality

Not everyone can be, or needs to be, a visionary or genius.  
The most important part of operations is being able to get things done.



# The Myth

“I have to work 80 hours a week just to keep up in this field.  
I don't have time for anything else.”



# The Reality

You shouldn't have to sacrifice your life, or your sanity, just to keep up in our field.



If you can't remember anything else...

...the worst that can happen is that you don't get the job.



# Positioning Yourself For Success



# Culture Shock



# A Ten Minute Walk From My Office

Then



Now





# Embrace the KISS Principle

Don't make things more complicated than you have to.



# Take a deep breath.

You can't fix everything -- not right away, anyway.



It's okay to be “boring.”



# You can fix things later if you have to.

An 80% solution today is better than a 99% solution in a month. Or six months. Or shortly after the heat death of the universe.



# Find Your Focus

Be a “T-shaped” engineer



# You don't have to be the best.

"Ninety percent of everything is crap."

You don't have to run faster than the lion, you just have to run faster than the other person.



# Learn the Business



Good technical decisions are good business decisions.





Don't live in a bubble



# Build your network

- Keep in touch with former coworkers (more than just LinkedIn)
- Conferences, like DevOpsDays and LISA
- Meetups
- CoffeeOps




# You'll never know if you don't ask.


hangops ▾  
● cblkwell


CHANNELS (211)  
# devopssparklyprincess  
# incident\_response  
# opsen  
# sfbayops  
# sre  
# \_hangops\_  
# aws  
# chef  
# gaming  
# job\_board  
# monitoring\_love  
# politics  
# puppet


DIRECT MESSAGES (3356)  
♥ slackbot  
● cblkwell (you)  
● charity  
● ed  
● geekgirl  
○ glasser  
● jamfish  
○ kcsch  
● kytrick  
● maplebad  
○ spinops


#incident\_response  
330 members | <https://github.com/Operations-Incident-Board> | See Pinned Items



**cblkwell** 11:24  
Are there any good articles folks can recommend on doing metaanalysis of postmortems/incident response?  
  
I'm working on a project now to try and improve our MTTD/MTTR and I'm trying to figure out what sort of metrics we want to pull out of incidents. I want to find a way to do it consistently and that will actually give us the answers we're looking for, but I'm wary of pigeonholing things in a way that will be counterproductive.  
  
Even just stuff like "how do you measure MTTD" -- I've heard of people measuring it from the time the code that presented the possibility for the incident was committed (to emphasize that testing/code reviews/etc should catch the issue) and others from the time it actually starts to have a site impact.

**becomingwisest** 11:46  
I'd say use the later normally, until you build up the before pipeline, and then have its own metric there

**cblkwell** 11:57  
Yeah, I'm coming around to the thought that going too far back makes determining the "start" of the incident too arbitrary and therefore less useful.  
  
And there's too much potential to pin the blame on the person who committed that line of code or whatever.

**jallspaw** 14:16  
[@cblkwell](#): I wouldn't measure it  
  
In fact I'd stay away from measuring it, because they're infinitely negotiable metrics

**cblkwell** 14:17  
Hrm...but doesn't Etsy measure that very thing in morgue?  
  
(which I was looking at this morning)  
  
You at least seem to measure TTD per-incident, which means MTTD should be relatively easy to determine same with MTTR  
  
You don't see value in measuring that? How can you tell if you're getting better at your incident response? That's really what I'm trying to answer here, I guess.  
  
We can try all these things, but if we can't measure it, it is hard to know if it is really improving outcomes.

**jallspaw** 14:20  
[@cblkwell](#): that's a capture of the constructed narrative. We don't do anything with it other than use it to help frame the debriefing  
  
<https://www.routledge.com/Safety-Cant-Be-Measured-An-Evidence-based-Approach-to-Improving-Risk/Townsend/p/book/9781409453116>  
  
 Routledge.com  
**Safety Can't Be Measured: An Evidence-based Approach to Improving Risk Reduction (Hardback) - Routledge**  
  
The British Prime Minister has vowed to 'bill off the health and safety culture' which he



Great places to look for answers:

- Hangops Slack (<https://signup.hangops.com>)
- Freenode IRC (<https://freenode.net>)
- StackOverflow (<https://stackoverflow.com>)
- Mailing Lists
- Twitter



# Build a library of resources

- SRE Weekly (<https://sreweekly.com/>)
- Yelp Engineering Blog (<https://engineeringblog.yelp.com/>)
- Etsy's Code As Craft Blog (<https://codeascraft.com/>)
- Arrested DevOps (<https://www.arresteddevops.com/>)



# Protect Yourself From Burnout

- Don't give in to the hero complex. Working too much is bad for you and bad for your company.
- Leave work at work, as much as you can.
- Take your vacation. Yes, all of it.



# So, to sum up...

- Resist the temptation to believe the myths about our industry and yourself.
- It's okay not to be the best at everything. Find something you're good at and do it well.
- Don't live in a bubble -- learn your business and stay in touch with the community.
- Protect yourself from burnout. Your job doesn't need to be your life.



# References and Inspiration

- <http://www.techrepublic.com/article/tech-industry-employment-soars-past-6-7-million-led-by-it-services/>
- <https://hbr.org/2015/08/the-research-is-clear-long-hours-backfire-for-people-and-for-companies>
- Davis, Jennifer, and Katherine Daniels. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale. Sebastopol, CA: O'Reilly, 2016. Print.
- A Young Lady's Illustrated Primer to Technical Decision Making, Charity Majors  
<https://www.usenix.org/conference/srecon16/program/presentation/majors>
- I Am A Sysadmin (And So Can You!), Ben Rockwood  
<https://www.usenix.org/conference/lisa14/conference-program/presentation/rockwood>
- Morrow Plots and TransAmerica Pyramid images courtesy Wikipedia under Creative Commons License







[fb.com/YelpEngineers](https://fb.com/YelpEngineers)



[@YelpEngineering](https://twitter.com/YelpEngineering)



[engineeringblog.yelp.com](https://engineeringblog.yelp.com)



[github.com/yelp](https://github.com/yelp)