

Migrating from VMs to K8s

We did it, and so can you!



Amazon
EC2



Kubernetes

KnowledgeLound

Who are we?

Allen Nelson

Software Engineer

- Full stack engineer
- Past DevOps experience
- Helped multiple companies transition their development models
- <https://www.linkedin.com/in/allen-nelson-05757050/>



Nick Petrovits

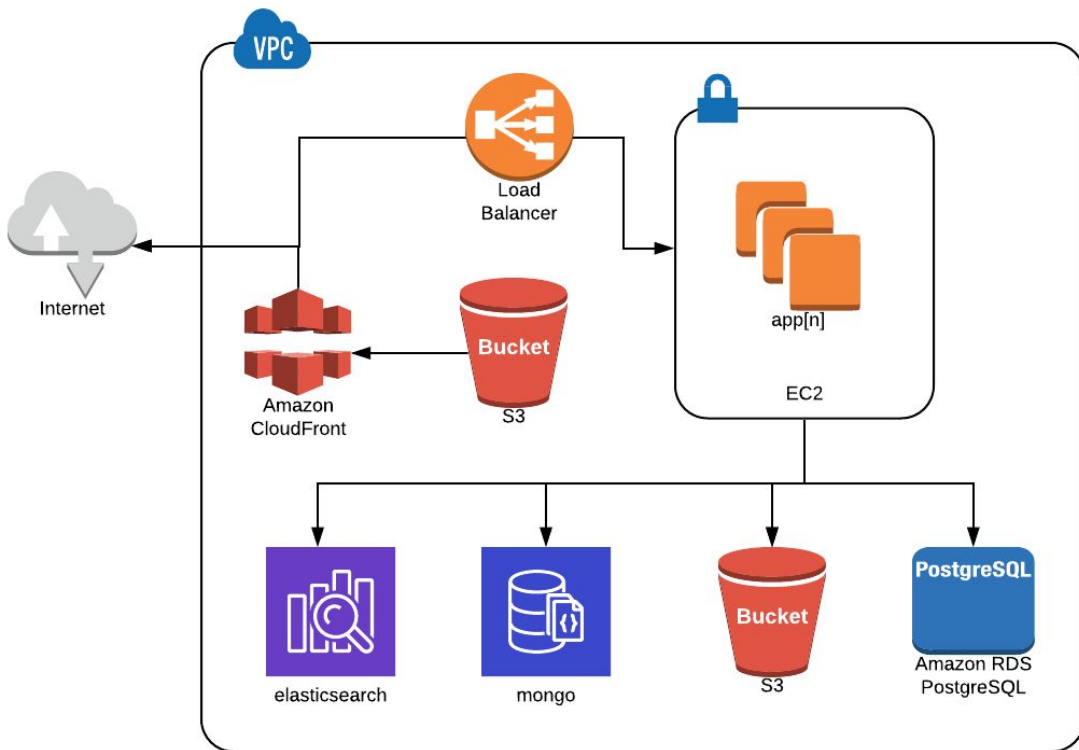
VP of Engineering

- Full stack engineer
- Using public clouds since 2011
- He has built multiple automated multiple deployment pipelines
- Led the automation and DataOps process optimization for large big data solutions
- <https://www.linkedin.com/in/nicholaspetrovits/>



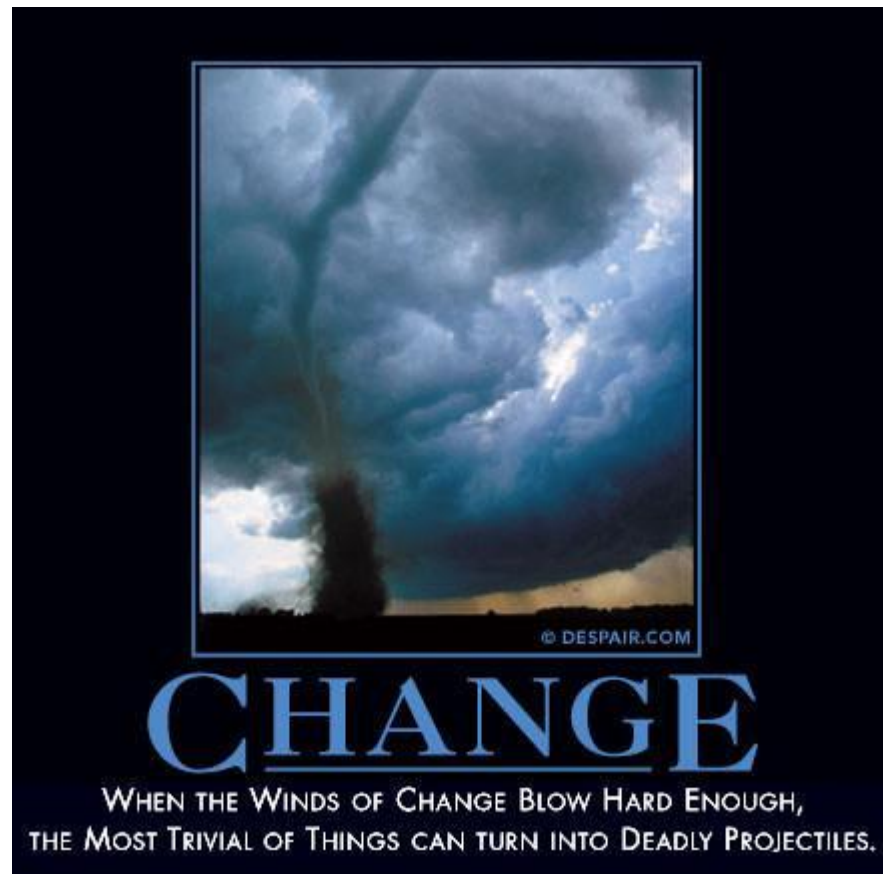
Where does this adventure start?

- A Load Balancer and EC2
- Deploying a few Django services to VMs is easy
- Updating dependencies is hard
- One service would never bring down both VMs!



We could try containers...

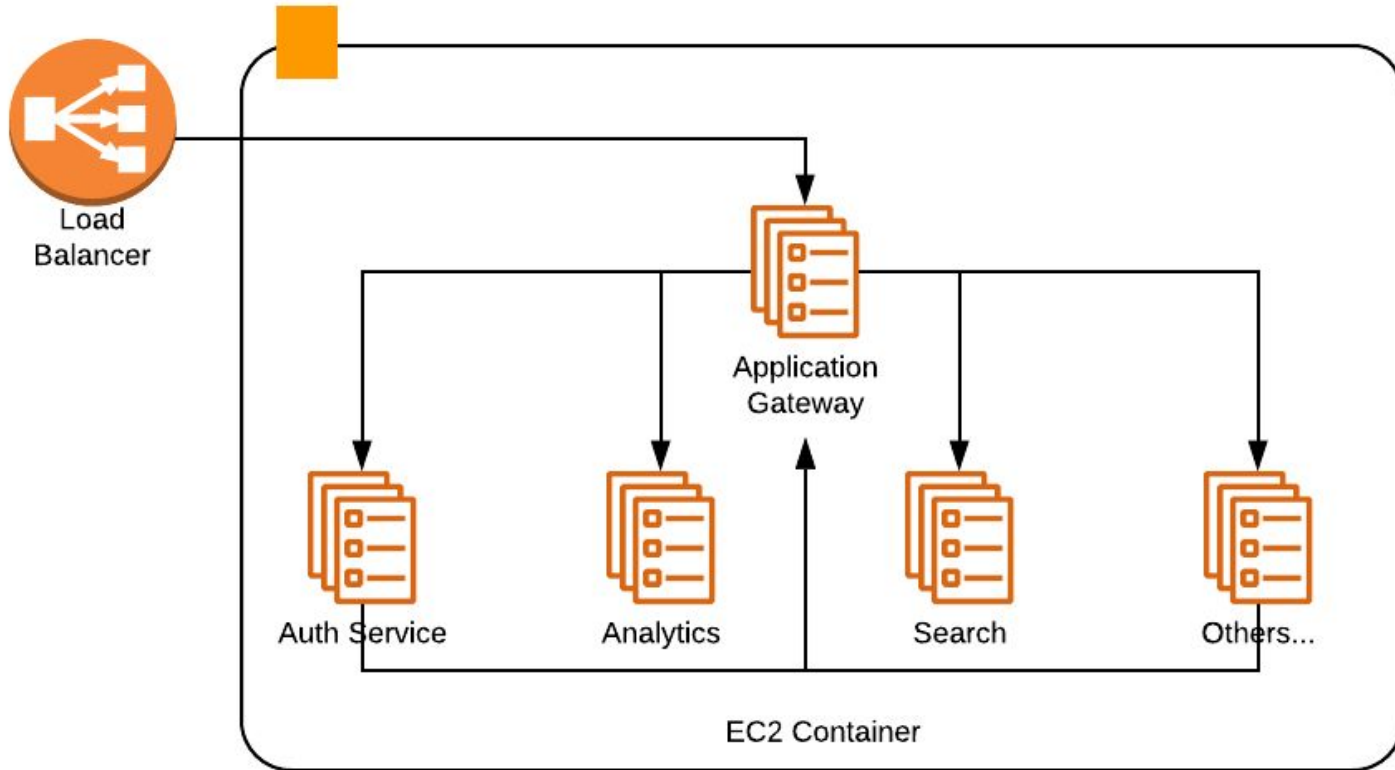
- Everything already works / K8s is hard
- Where do we start?
- No DevOps Team
- How do we stop everything and migrate?
- We are a small startup and this is a large opportunity cost!
- Murphy's Law - A single service won't crash the entire site again!
- Let's just go serverless!



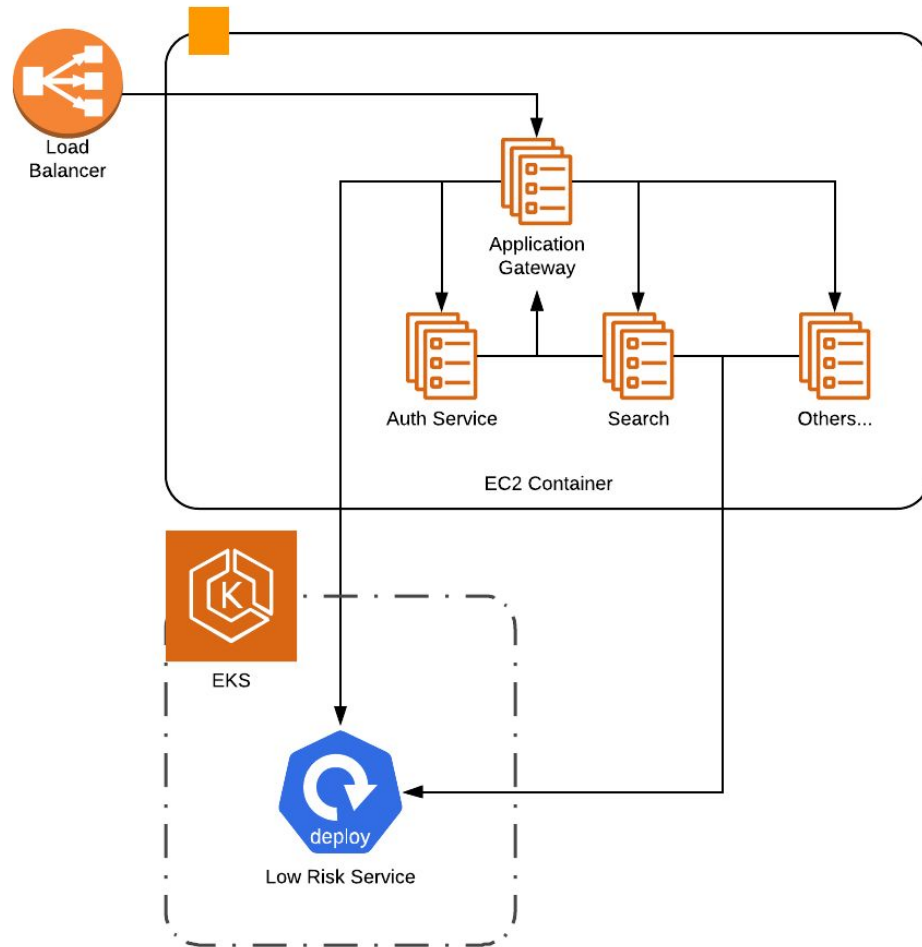
**YES,
WE
CAN.**



Where do we start again?



Step one!



What to transition with first?

- Small
- Non-critical
- Stateless
- New
- Representative



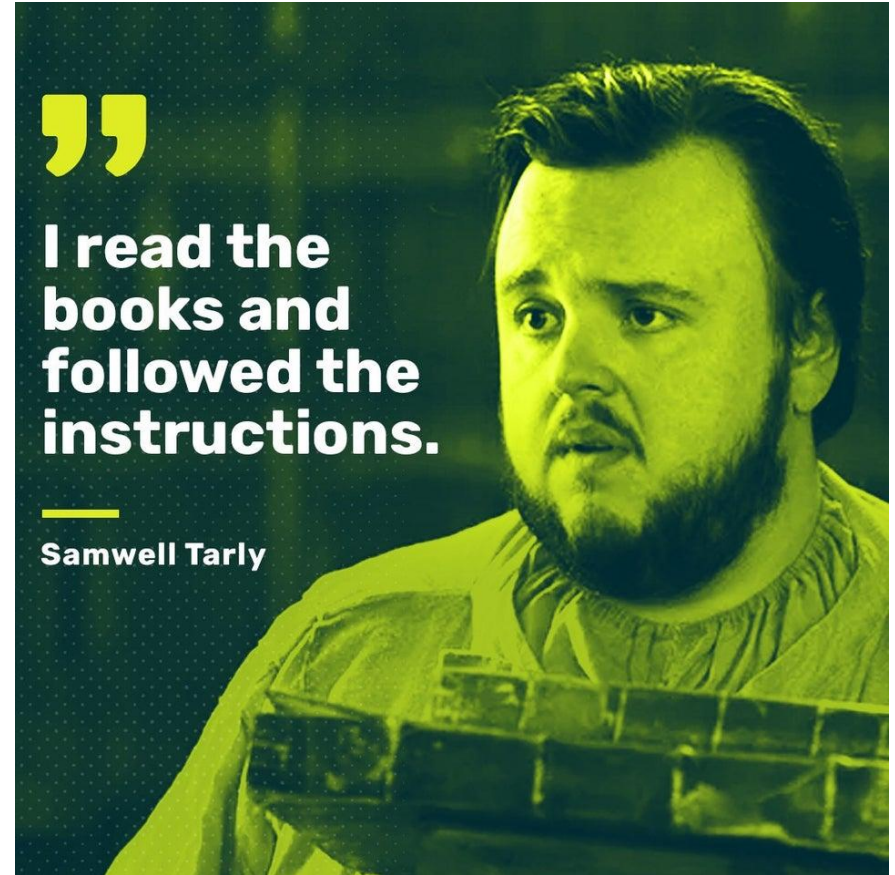
Don't piss off the rest of the team

- Script everything
- Dark deploy
- Avoid long-separated branches, merge/rebase often
- Make it easy to flip between versions and avoid downtime in environments
- Create a “1 Pager” with common dev, troubleshooting, and deployment tasks
- Teach the dev team before deploying



Documentation

- Detailed, step by step documentation
- High-level, quick notes documentation



How do we develop now?



- Local infrastructure managed through `vagrant` and `docker-compose`
- Configuration, scripts and infrastructure documentation live in a repository named `devops`
- One repo per service
- Developer is responsible for running services

I just want to deploy my code



Devops scripts can be awesome

```
$ k8s-deploy SERVICE --to int  
$ k8s-deploy SERVICE --to prod --tag TAG  
$ kh-pods int SERVICE --ssh  
$ kubetail SERVICE
```

Devops scripts can be terrible

```
$ kh-swiss-army-knife deploy --run-tests  
--notify-slack --type k8s --replicas 6  
--max-memory 10GB --branch my-branch
```

Lots of things happen...

Error: Some step failed and now you have no
idea what to do

```
$ which tail-my-service  
kubetail --some-option my-service
```

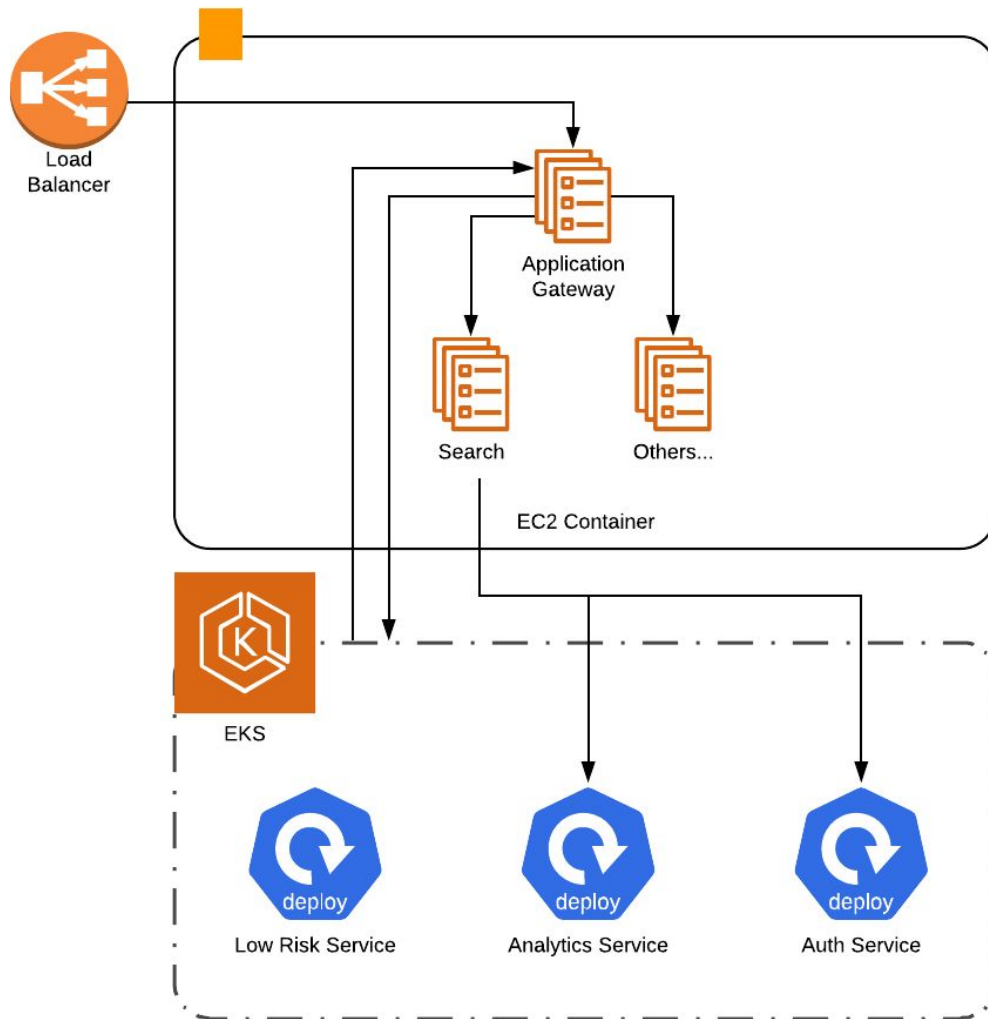

Networking



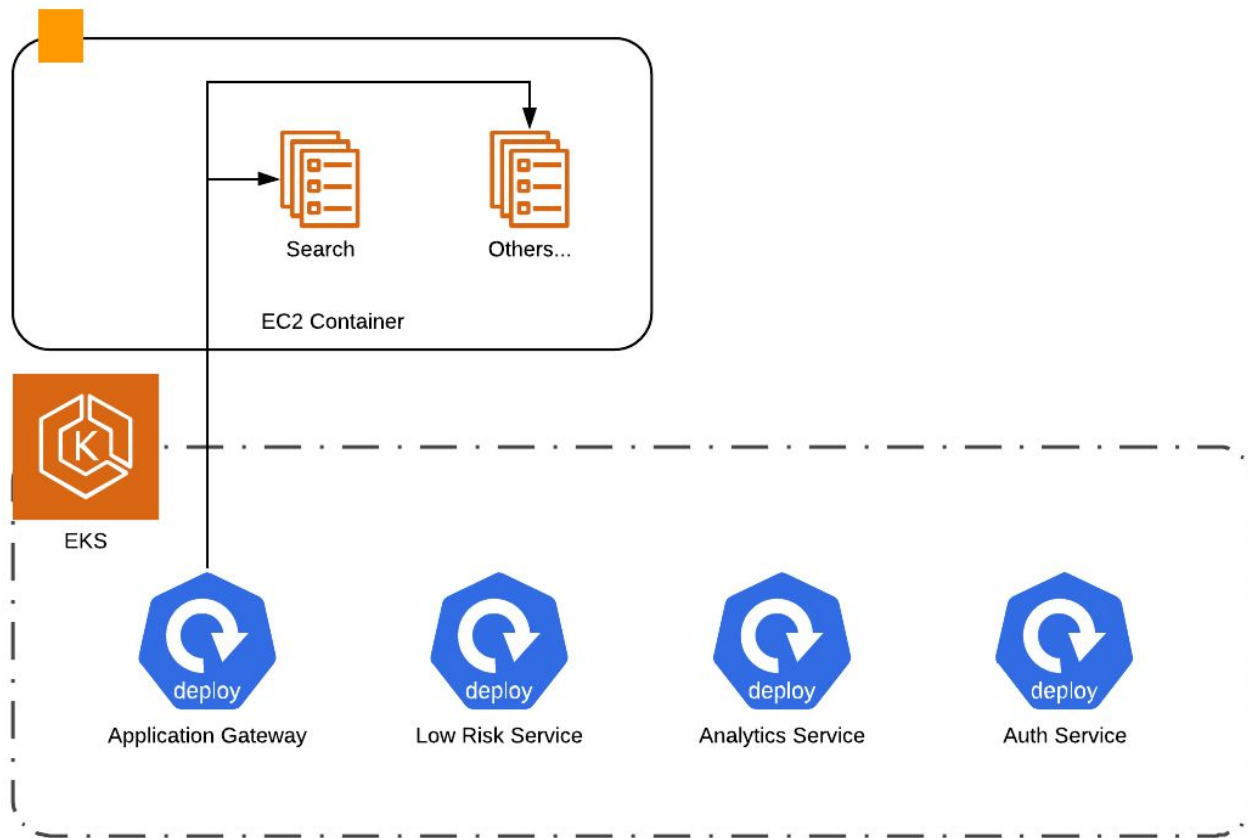
```
# cat /etc/hosts/apache2/myserver.conf
<VirtualHost *:1234>
    ServerName myserver

    <Location "/">
        ProxyPass "http://10.0.2.2:1234/"
        ProxyPreserveHost On
    </Location>
</VirtualHost>
```


We can do this!

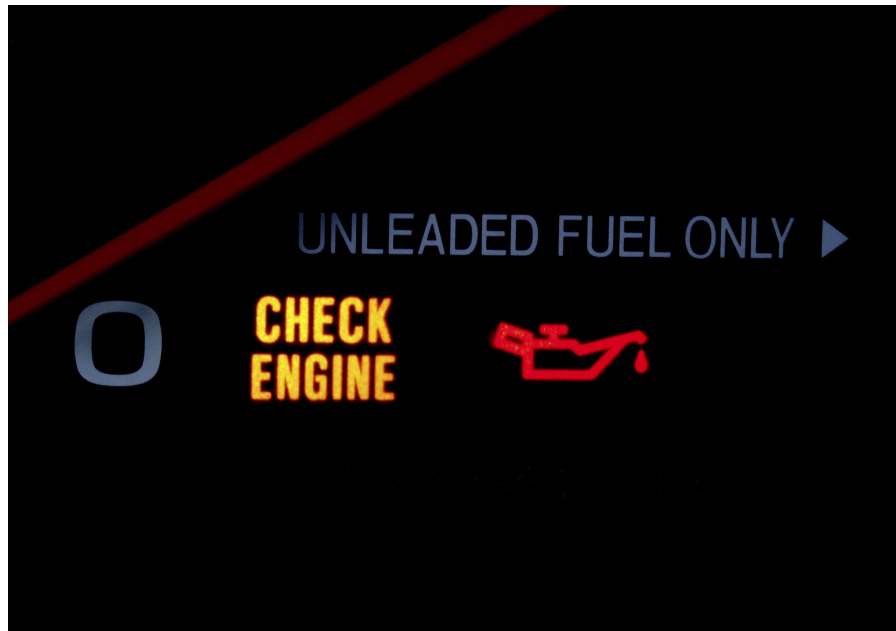


What's Next?

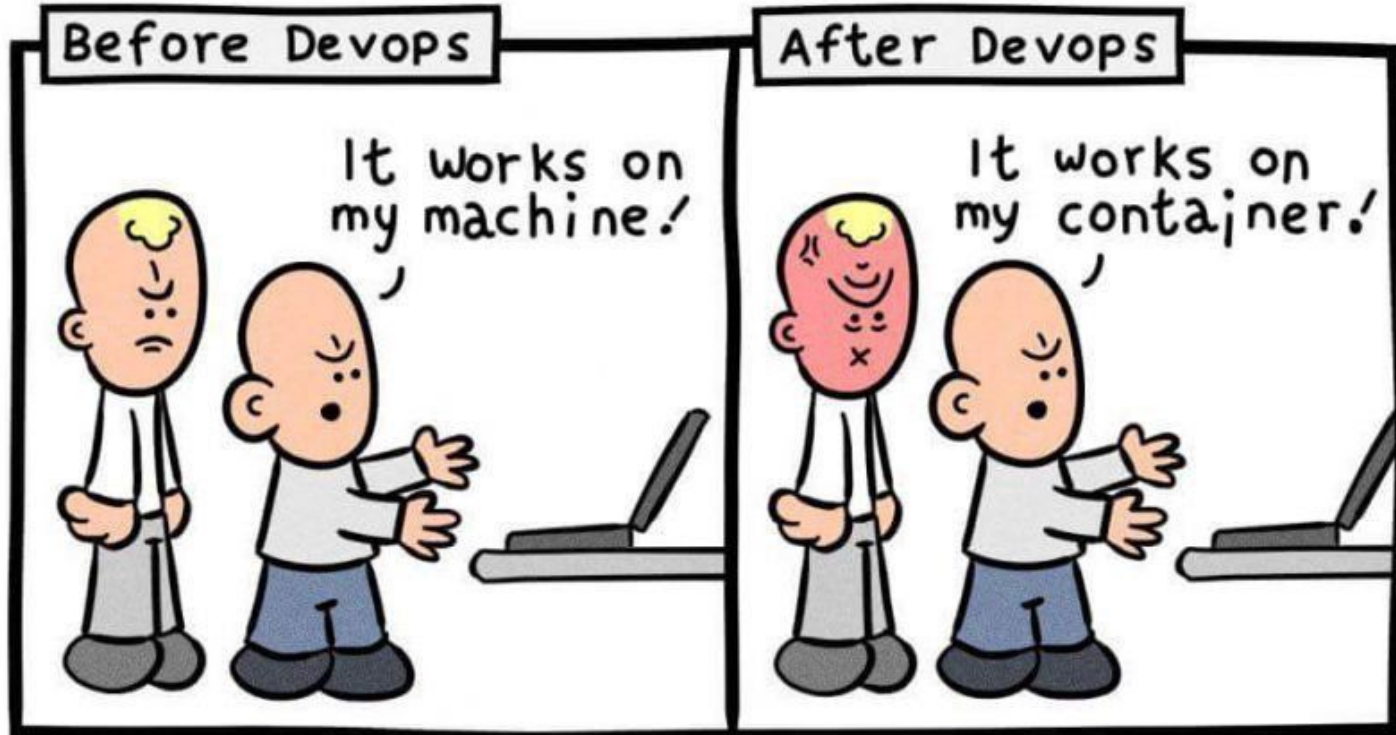


Resource Usage and Stability

- You don't need to guess the exact resource requirements off the bat
- Start with a reasonable estimate, and then revise as needed observing production metrics
- Redundancy/replication is easy in Kubernetes. Use it!

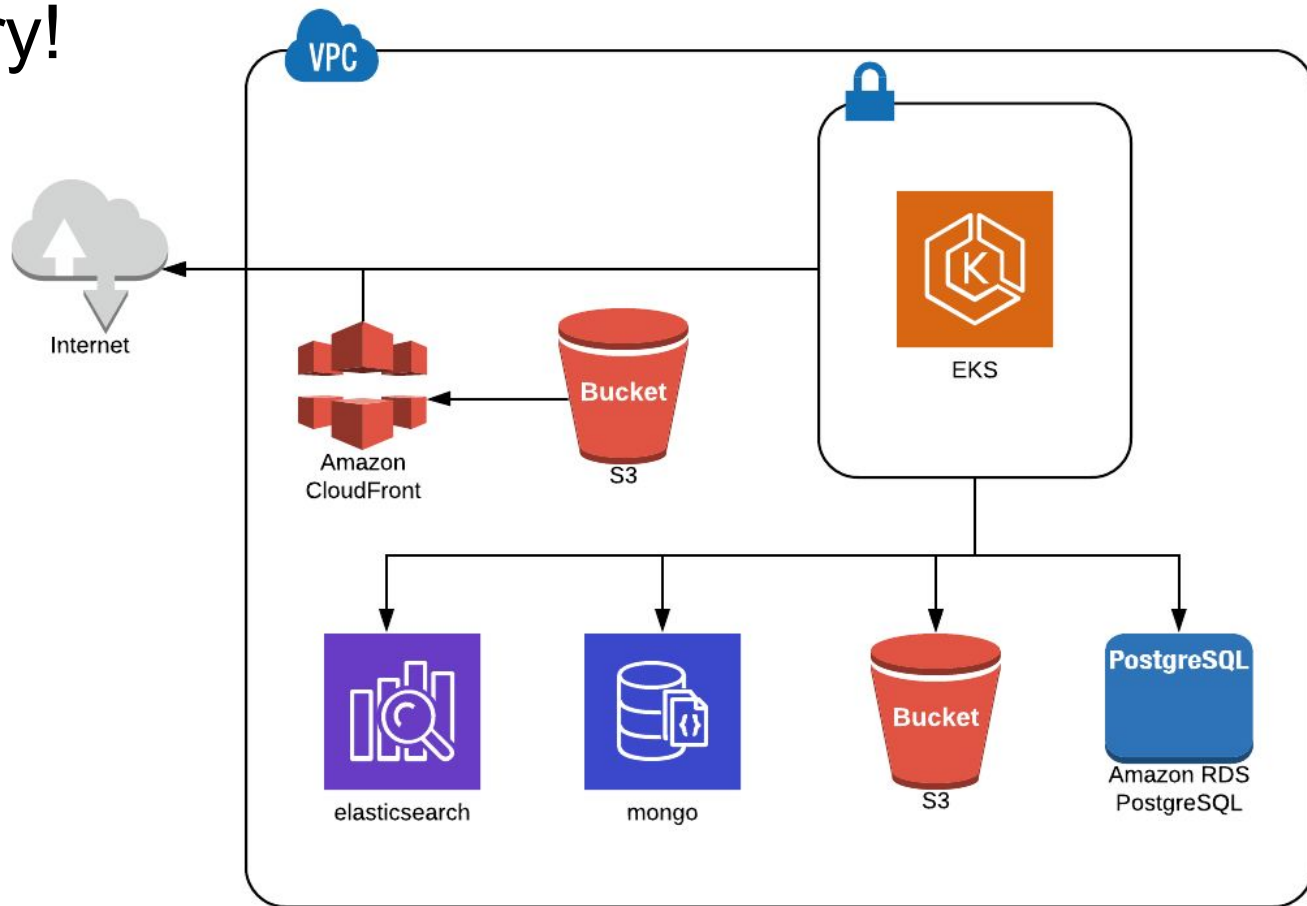


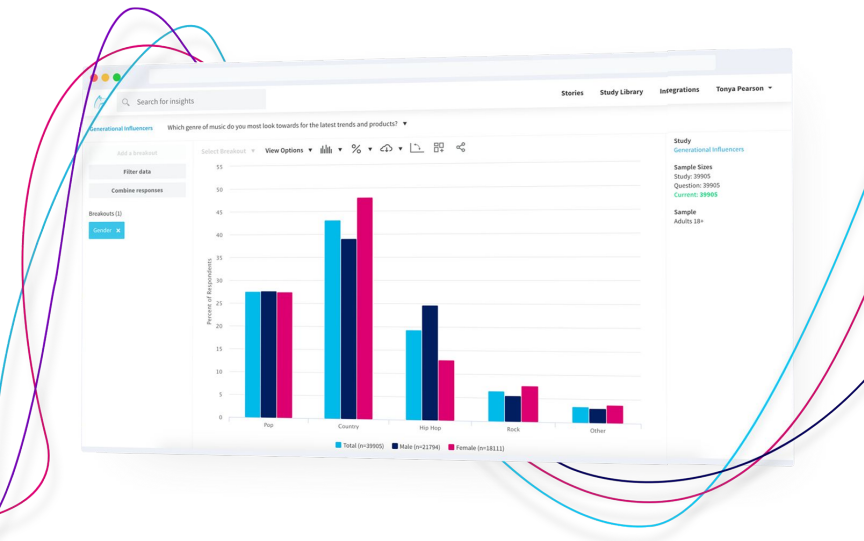
CI/CD



Daniel Stori {turnoff.us}

Victory!





We make teams more customer driven.

Through an insight
discovery and data
visualization platform.

