

*I keep hearing customers  
wondering whether or not  
they even need  
kubernetes*

— @ericlongtx

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

So in january I was having dinner with some folks when this came up

Slide 2

**2018 CALLED  
IT WANTS ITS  
EPIPHANY  
BACK**

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

And all I could think of was this

# You might not need Kubernetes

Monday, November 5, 2018

Jessie Frazelle

@jessfraz

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

Jessie Frazelle was one of the first people I remember seeing state this explicitly

Read Jessie's posts  
**that's it**  
that's the whole talk

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

And I know people have to have a hot take, so here's mine

# ok here's more: First, Principles

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

But that doesn't make a good ignite talk, so let's talk about why and why not k8s

**Kubernetes is  
no good  
for fixing your culture**

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

Your applications depend on people working together effectively

**Kubernetes is  
no good  
for orgs not already measuring**

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

Applications on kubernetes often pull in a world of network-heavy abstraction layers, which can really screw with your expectations

# Kubernetes is good for strangling legacy apps

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

My opinion is that using a move to k8s to strangle old code can prove to be one of the best reasons to switch, but you've gotta be ready



**Kubernetes  
is good for  
experimenting in prod**

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

The tooling for writing distributed scalable apps on k8s can be awesome

# How are those abstraction layers treating you now?

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

Are you already using layered discovery and injected config?

Are you already using containers?

# What's working for you now?

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

Or are you living in a golden-image-based vm world with tons of change management burden?

How many groups need to weigh in to approve a platform-level configuration change

# Complexity Hot Potato Is No Fun

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

If you think k8s will hide complexity from your devs glhf

**Debugging  
is twice as hard  
as writing a program  
in the first place**

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

Layers of abstraction can mean figuring out what's breaking your app awful, particularly if you don't know how to build a platform

**You didn't  
write k8s  
good luckin prod**

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

All this doesn't even cover running k8s as a platform, if you aren't gonna buy managed k8s from a cloud provider.

**Then again  
you didn't write  
vSphere**

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

You probably already learned plenty of these lessons on vmware, openstack, BYO bare metal

# Sufficiently Advanced Technology *is* Magic

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

If you figured out how to make humans work together to make those platforms work for you, you've already got some of the muscles you'll need to do a k8s to your stuff reasonably well



# Kubernetes doesn't matter (to your users)

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

Remember what actually matters

Make sure it works for your audience

Whether that's developers or people wanting to upload dog pictures to your dog park playdate app

# Use What Works

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

Please don't buy into the hype  
Start simple and add layers when they make sense, after talking to people further along in their journey  
OODA your way to happiness

# Own Your Availability

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

This is always your responsibility

Don't play the blame game

Your app doesn't work just because it's on ec2 or break because it's on k8s

**Thanks**  
I've been [@sdmouton](#)

 [@sdmouton](#)

Alrighty, that's my time, hugs