

Why developers  
shouldn't care about  
containers



Why developers

- ~~shouldn't~~ needn't care  
about containers



Hello!

**I AM ED SHEE**

Cloud Developer Advocate at IBM

You can find me at [@ukcloudman](#)

Imagine it's early 2014...

Docker is fairly new but is gaining popularity

You put some code in a container...



But your container isn't much use in production alone...



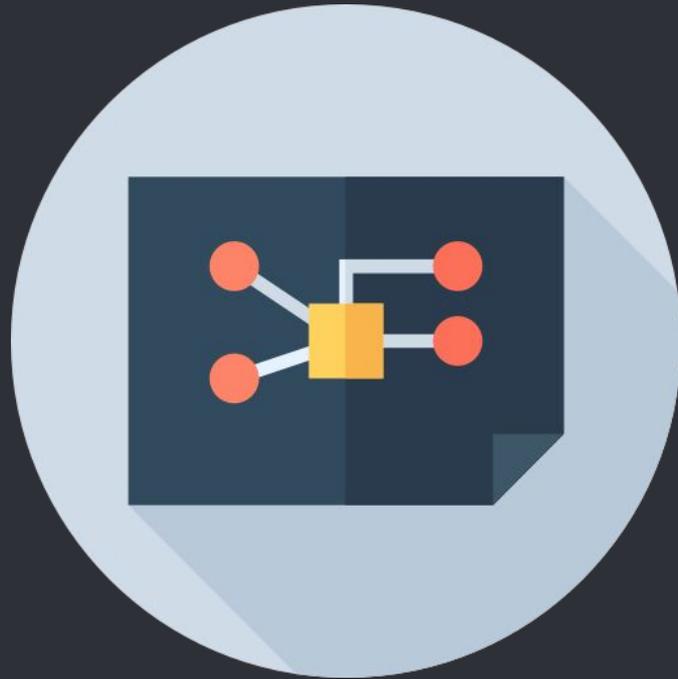
# Automated Deployment



# Scaling



# Routing



# Logging



# Monitoring



# Database



# Service Discovery



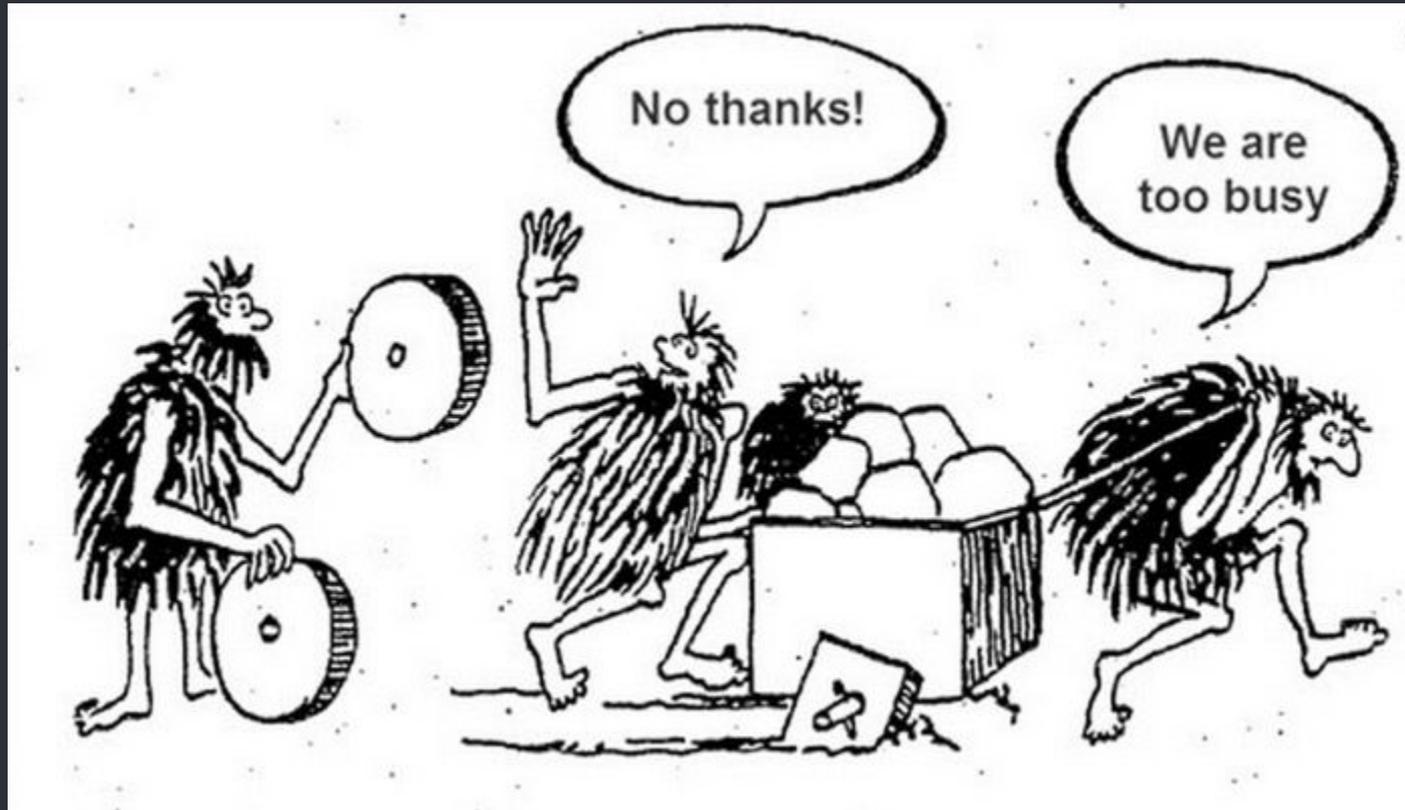
Suddenly your container isn't so lonely



Setting all of this up takes a lot of  
time!

Even though there are fantastic devops tools

Don't be like this...



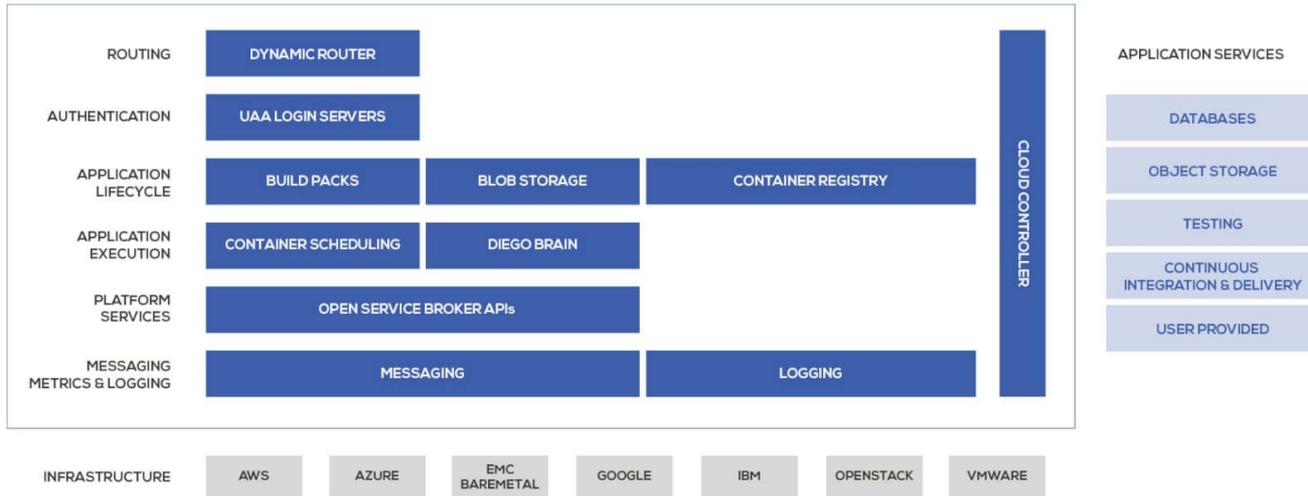


**Use a platform that already exists!**

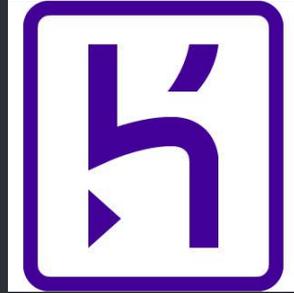
# Cloud Foundry Application Runtime



## PLATFORM ARCHITECTURE



## PaaS Providers





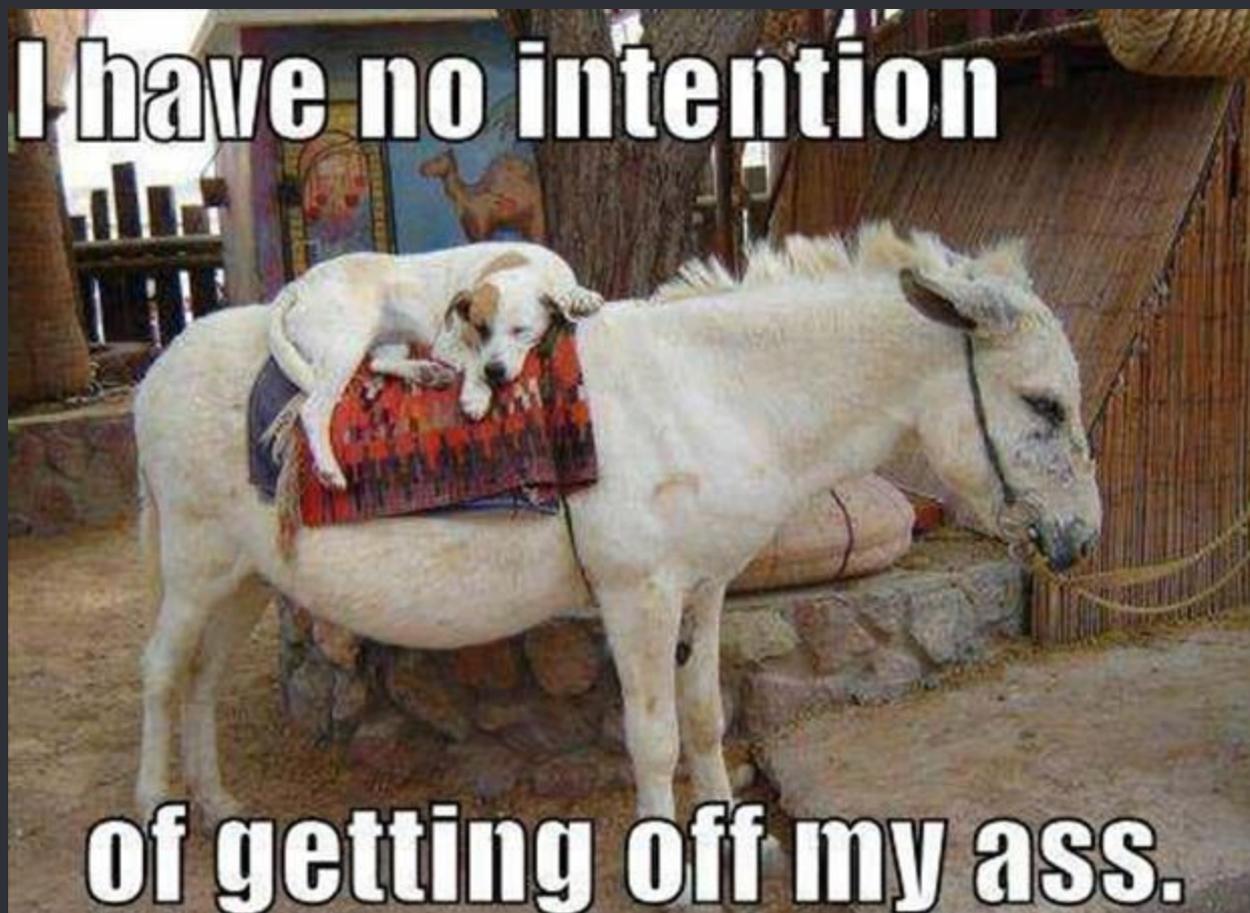
**Demo**



**COOL STORY, BRO**

## What about all the other stuff my app needs?

- Databases
- Message Queues
- User Directories
- Security Services
- Machine Learning Services



**I have no intention**

**of getting off my ass.**



**Focus on actually developing the software!**

# What about Serverless?

Serverless is the future...

# Serverless is Dead

What we should instead be focusing on is what we're seeing to be the new way of doing modern application development

- Greatly reduced operational burden
- Tighter alignment to costs w/ usage
- Developers can/could/should be able to do almost anything
  - understated warning to Ops/DevOps/SRE/-ish folks
- Opinionated platforms that allow for multiple use-cases
- Opinionated platforms that bake in true best of breed practices, security, scale, performance, cost aspects for you

Chris Munns, Principal Developer Advocate - Serverless @ Amazon Web Services

# What about Kubernetes?

I can do all that and more with Kubernetes...



**Kelsey Hightower**

@kelseyhightower

Following



Perception: I'm using pure Kubernetes; I don't need a platform.

Reality: Everything you do above kubectl is proof you need a platform and you're actually building one.

9:43 AM - 24 Feb 2019

365 Retweets 1,190 Likes



37



365



1.2K





**Kelsey Hightower** ✓

@kelseyhightower

Following



Kubernetes is a huge step up from IaaS and provides a ton of value. It's a bottoms up approach to platform building and offers more flexibility than fully opinionated PaaS offerings.

9:59 AM - 24 Feb 2019

52 Retweets 220 Likes



4



52



220



Tweet your reply



**Kelsey Hightower** ✓ @kelseyhightower · Feb 24



The delta between Kubernetes and a developer friendly PaaS is where the next layer of value is and where things tend to get opinionated -- a requirement for reliable end to end workflows.



6



26



122



**Kelsey Hightower** ✓ @kelseyhightower · Feb 24



While all this platform building is happening the question you have to ask yourself is: do I need to build, run, and manage my own platform?



16



25



115



# Kubernetes

## Container Centric

Allows complete flexibility and control over the container environment. It is a **platform for building platforms**.

Cloud Foundry

## Application Centric

Easier to learn, quicker to use but comes at the cost of flexibility.

**To be simple it needs to be opinionated.**



**So who should care about containers?**

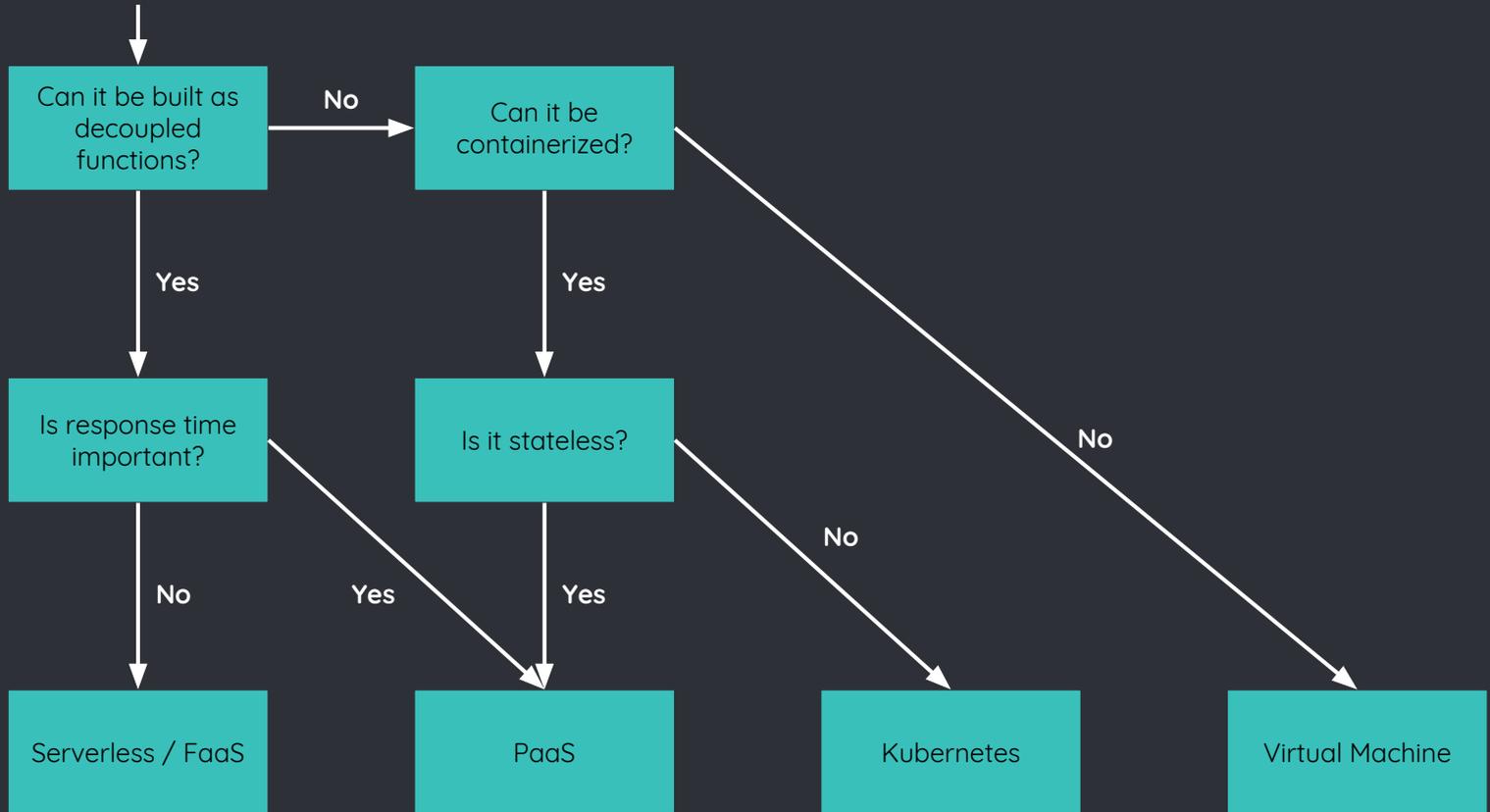
In depth container knowledge is perfect for:

- DevOps engineers (at large companies)
- Building a Cloud Platform
- Running a highly available distributed service
- Deploying highly customised workloads
- Enthusiasts!



Where should I put my workload?

# Start building a new app...



## Summary

- Abstract as much as possible
- Rent/buy as much as possible
- Focus on software development
- Learn containers only if it makes sense to do so

Thanks!

**ANY QUESTIONS?**

You can find me at

@ukcloudman

edmundshee@uk.ibm.com