Kubernetes Is Not For Developers
And Other Things the Hype Never Told You
Taylor and Jordan

Intel / Puppet -> Nike -> MS
App Dev
Why Bother?

➔ Kubernetes abstractions (knobs) are great!
➔ Deploy quickly and safely
➔ Self-healing infra
Platform Dev
Mind the Gap

Responsible Kubernetes users limit their own options by abstracting away what is constant within their organization.
Observe their current ops practices
Learn about their apps
Identify their business requirements
Most importantly: Figure out how they think about their systems
Software Time
Scripting

➔ Python/Bash/Ruby/etc.
➔ Very flexible
➔ Not already done for you
➔ More difficult to maintain
Helm

➔ Provides managed “releases” of an app
➔ Basic lifecycle management
➔ Has a structured user input system (the values file)
➔ Currently requires a server side component
CRD + Controller

➔ Reduces your use of Kubernetes to a single resource type
➔ Users may still need to interact with kubectl
➔ Libraries exist to help build controllers
➔ Coupled to Kubernetes
Custom API

- Provides a minimal, opinionated interface for managing apps
- Users don’t have to worry about Kubernetes at all
- HTTP is a good delineation point between services
- Isn’t good for non-homogeneous workloads
- Extra knobs can quickly become unwieldy
Questions?
Thanks!

Art by emilyolshevski.tumblr.com