

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 are great!

Platform Dev



Mind the Gap

Responsible Kubernetes users limit their own options by abstracting away what is constant within their organization



Nitty-Gritty

- Observe their current ops practices
- Learn about their apps
- Identify their business requirements
- Most importantly: Figure out how they think about their systems

DetailsImpl

- Service/Ingress
- RBAC/{Pod,Network} Policy
- Deployment/Horizontal Pod Autoscaler
- Limit Ranger

Software Time

Scripting

App Dev

- Invokes script with arguments from dev workstation or the “script runner”

Platform Dev

- Implements script and defines arguments
- Chooses the “script runner” (e.g. Jenkins, Concourse, CircleCI)

Helm

App Dev

- Manages releases and associated values
- Gets state of the app with `kubectl`

Platform Dev

- Writes and maintains chart

CRD + Controller

App Dev

- Manages application by instantiating and modifying resource

Platform Dev

- Builds controller and custom resource spec

Custom API

App Dev

- Calls API to create application
- Gives some sort of artifact
(container, binary, etc)

Platform Dev

- Creates and defines API
- Writes the “controllers” or
business logic for running the
app

Questions?



Thanks!

Art by emilyolshevski.tumblr.com