

My Life In DevOps

Two years of roaming the AWS landscape

Northwestern

Michael B. Klein, *Development Tech Lead*

David Schober, *Development Team Lead*

Northwestern University Libraries

Repository & Digital Curation Workgroup

Samvera Connect 2018

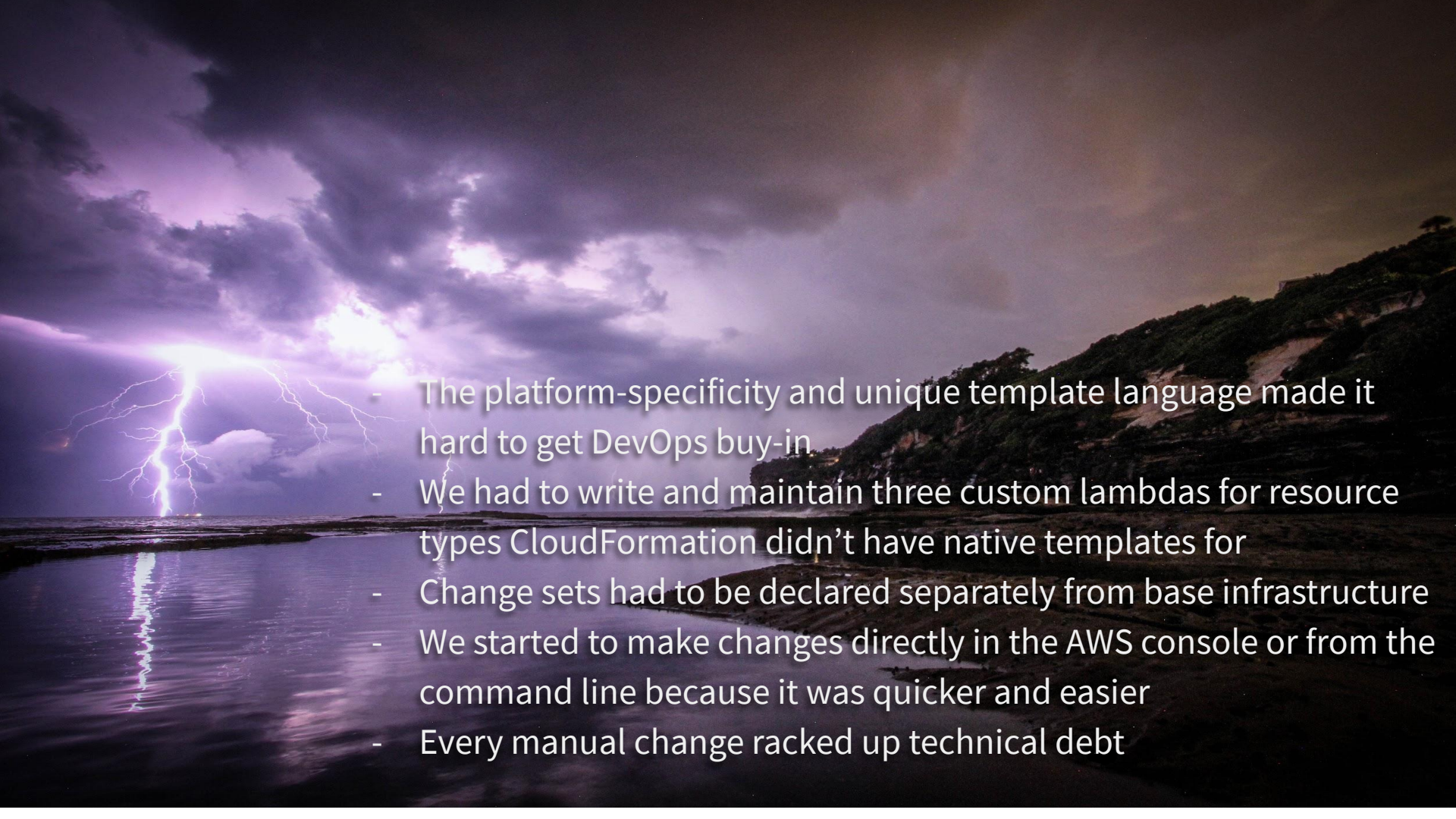
Well, MBK's life in DevOps



Prologue: When last we met

- We adapted Hyku's CloudFormation templates to suit our needs (thanks!)
- We separated out the stack from the applications to more easily allow for multiple heads
- We tacked on more AWS-native services
- We kicked back and started making mojitos



- 
- The platform-specificity and unique template language made it hard to get DevOps buy-in
 - We had to write and maintain three custom lambdas for resource types CloudFormation didn't have native templates for
 - Change sets had to be declared separately from base infrastructure
 - We started to make changes directly in the AWS console or from the command line because it was quicker and easier
 - Every manual change racked up technical debt

Our Infrastructure Got Complicated


- Three different Elastic Beanstalk solution stacks:
 - Fedora (Java)
 - SOLR (Multicontainer Docker)
 - Zookeeper (Multicontainer Docker)
 - Heads: Hyrax x2, Avalon (Ruby/Puma)
- Different defaults, deployment requirements, and options for every app on every stack.



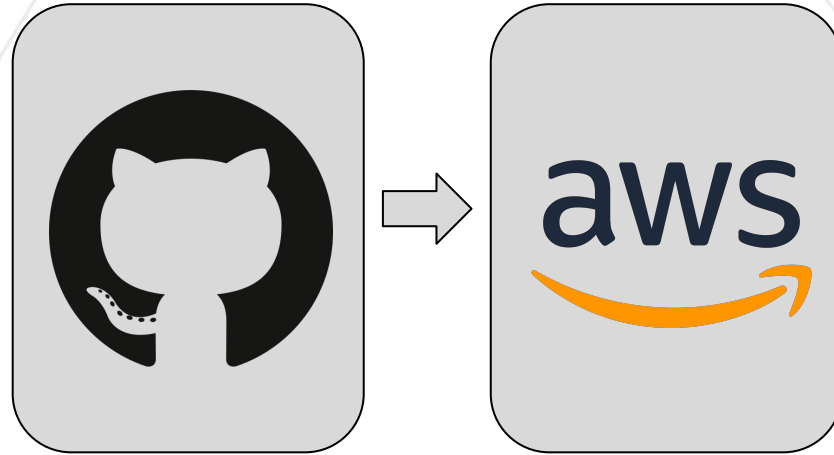
We started fresh with Terraform...

- Granular control of AWS resources
- Openly and actively maintained
- Tons of community modules
- Our Puppet-trained DevOps team seemed happier
- Supports multiple providers (Google Cloud, Azure, etc) so we have an escape hatch

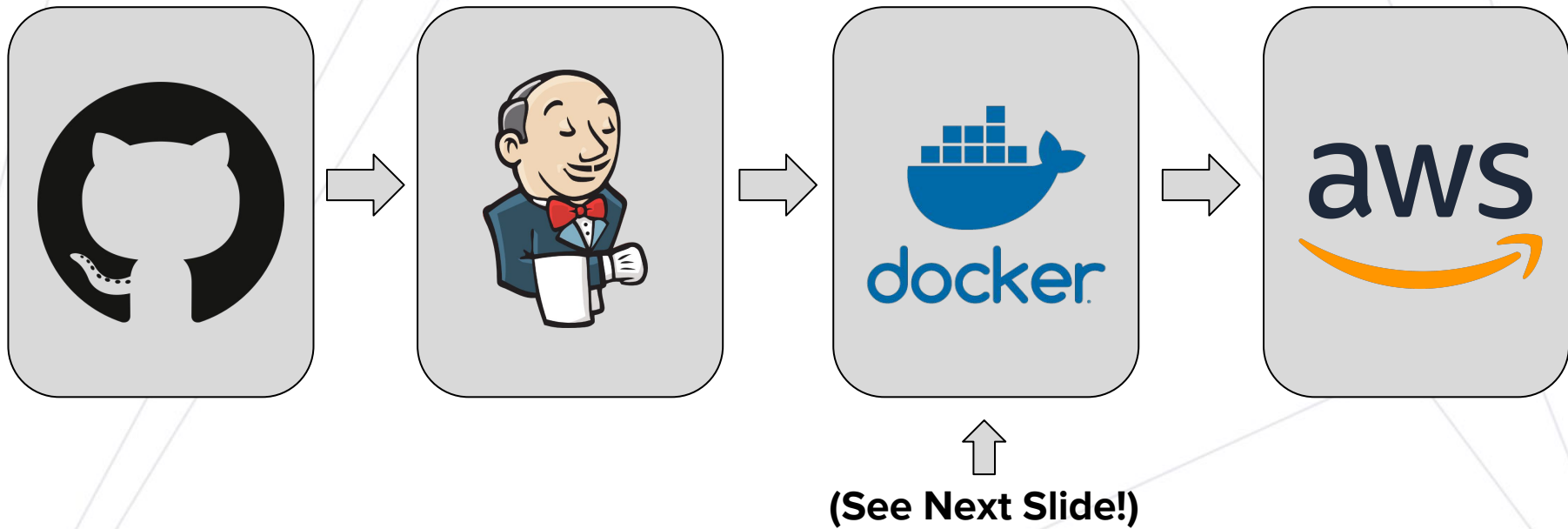
...and used it as an opportunity to re-re-evaluate

- 
- Use managed AWS services when possible (Redis, Database, SQS, Elastic transcoder)
 - Use containers for *everything* else
 - We're still thinking about:
 - Fine control of Elastic Container Service vs convenience of Elastic Beanstalk
 - Fargate vs EC2 in Elastic Container Service
 - And, heck, should we look at Kubernetes/EKS while we're at it?

Deployments Then



Deployments Now



Single-Stage Build

```
FROM ruby:2.4.4-slim-jessie

RUN apt-get install -y $BUILD_DEPS

RUN # Download & Install FFMPEG

RUN # Download & Install FITS

RUN # Download & Install VIPS

COPY . /home/app/current/

RUN chown -R app:app /home/app/current && \
    bundle install --path vendor/gems" && \
    rm -rf vendor/gems/ruby/*/cache/*

RUN bundle exec rake assets:precompile

EXPOSE 3000

CMD bin/boot_container

HEALTHCHECK --start-period=60s \
    CMD curl -f http://localhost:3000/
```

Multi-Stage Build

```
FROM ruby:2.4.4-slim-jessie as base

RUN apt-get install -y $BUILD_DEPS

RUN # Download & Install FFMPEG

RUN # Download & Install FITS

COPY Gemfile /home/app/current/

COPY Gemfile.lock /home/app/current/

RUN chown -R app:app /home/app/current && \
    bundle install --path vendor/gems" && \
    rm -rf vendor/gems/ruby/*/cache/*
```

```
FROM ruby:2.4.4-slim-jessie

RUN apt-get install -y $RUNTIME_DEPS

COPY --from=base /ffmpeg/* /usr/local/bin/

COPY --from=base /fits/* /usr/local/fits

RUN # Download & Install VIPS

COPY . /home/app/current/

COPY --from=base vendor/gems vendor/gems

RUN bundle exec rake assets:precompile

EXPOSE 3000

CMD bin/boot_container

HEALTHCHECK --start-period=60s \
    CMD curl -f http://localhost:3000/
```

BONUS! Developer Environment Got Better

- As close to a real mirror of production as possible
- Containers for everything
 - Same Cantaloupe, Fedora, & Solr that we use in production
 - 3rd party containers for AWS native services
- Single command startup
 - devstack up
- Multiple environments
 - Debug in a clean environment without losing existing development data



Hard things [WIP]

- Still working through Docker and AWS best practices (there's a lot of docs)
- Stuff broke, now what
- Docker containers aren't VMs which is great, but Docker containers aren't VMs

Use our stuff!

Most of these resources are pretty Northwestern-specific at this point, but adaptable.

- Terraform Repo: <https://github.com/nulib/nulterra>
- Docker images: <https://hub.docker.com/u/nulib>
- Devstack: <https://github.com/nulib/devstack>
- We try (but don't always succeed) to release well-documented code

Contact us!

- Michael B. Klein (Development Tech Lead, Repository & Digital Curation)
Email: michael.klein@northwestern.edu / Samvera Slack: @mbklein
- David Schober (Development Team Lead, Repository & Digital Curation)
Email: david.schober@northwestern.edu / Samvera Slack: @david.schober