Esri Developer Summit

March 8–11, 2016 | Palm Springs, CA

ArcGIS Open Data: DevOps

Daniel Fenton

TL;DR

- Build your infrastructure with code
  - Version controlled
  - Replicable
  - Automated
  - Transparent
- DevOps is a team responsibility
- Focus on workflows
- Optimize for getting to production, not to dev/qa

Background on our app

- Connected to but independent from ArcGIS Online
- Communicate only via publicly accessible APIs
- Built by a R&D team -> Goal is larger than the product itself
- Initial plan was for no backend at all

Beta launch

- Monolithic
  - Rails API
  - Backbone front-end for admin and consumer apps
  - Ruby workers for harvesting
  - Postgres for data management and search
- Opsworks + Chef
- Big jump in management from the original plan of no backend!

The start of decoupling

- Serving downloads from a sync rails app not so good
- Intro Koop
  - Node.js App
  - Traffic routed thru rails
- Decoupled enough for a start
- But, infrastructure is all managed by one guy
  - He’s still writing some back-end code

Initial launch

- Figured out Postgres was no good for search
  - Added ElasticSearch
- Still able to move quickly
  - Not that much traffic or customers yet so strain was low
- Made good use of feature flags

Liberating the front end

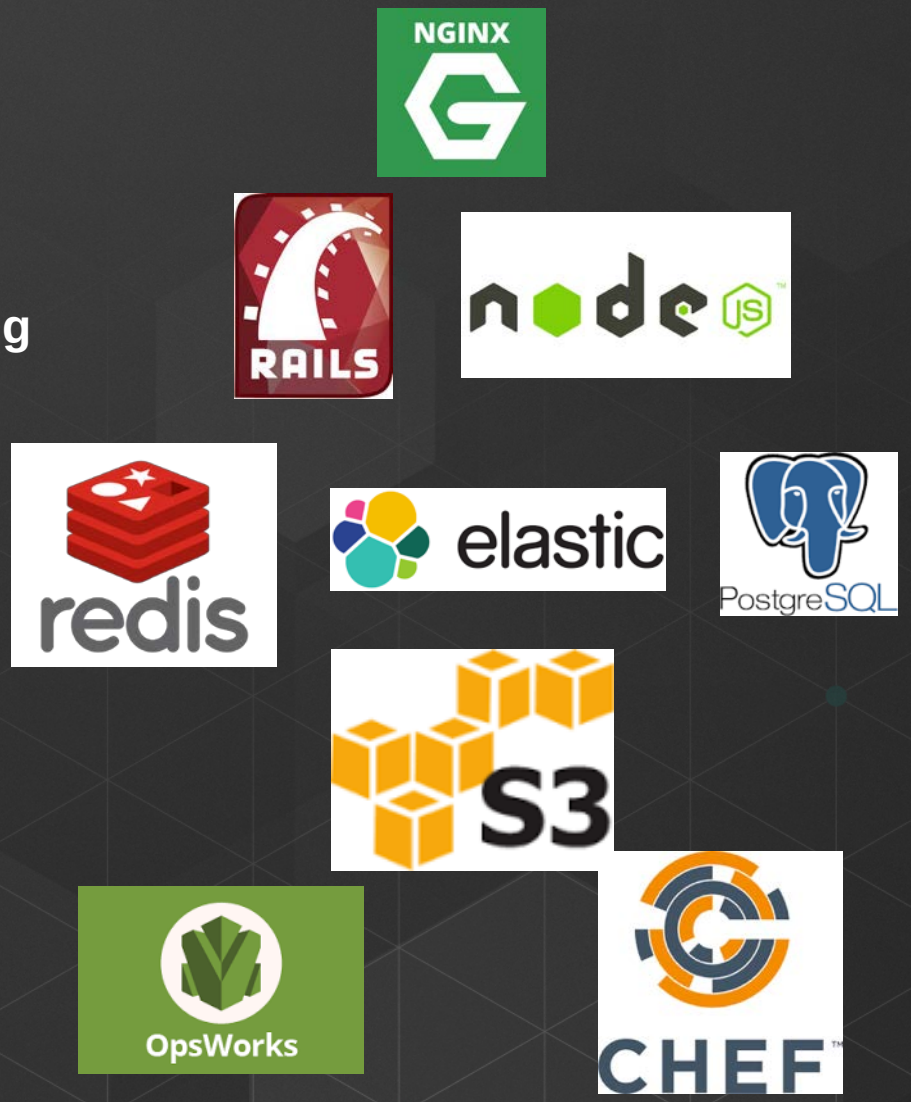
- Front end guys wanted to move faster
- Just chucking static assets onto S3
  - Still though, several new things to manage on AWS
- Push button deploy
- Dom for customer sites is still cooked by Rails app

Struggles with our Ruby Workers -> Decoupling to the Rescue (Again)

- Ruby harvesters were having trouble keeping up with the demand
- Major refactor didn’t solve the problem
- Decoupling and Node.js to the rescue!
  - Communicate via API with Rails
- Stack is pretty complex on it’s own
  - Node API
  - Postgres DB
  - Redis cache/message broker
  - Node workers

Current affairs

- Four functional groups:
  - Front End, API + Search, Downloads, Harvesting
- Everything stitched together via an NGINX Cluster
- Replicated across 3 stacks
- Complex networking
- Lots of snowflakes



Ouch!

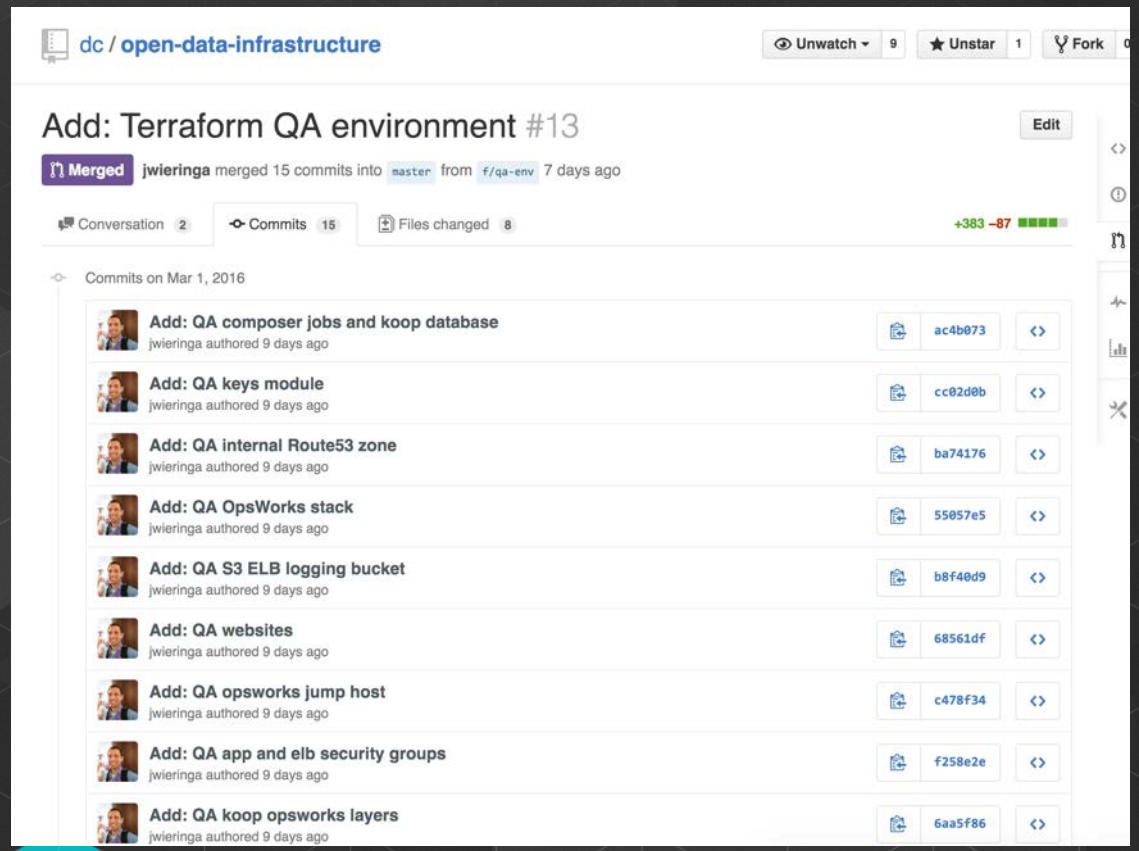
- Configuration of the entire stack by hand across 3 environments
  - Drift across the 3 envs
  - Regressions and exciting new bugs in production
  - Enormous amount of time to make even small changes
  - Forget about adding new things to the stack
  - QA process can take 8 hours for smoke test suite
- Final straw was 3 false starts on building a centralized logging system

Enter Terraform for Infrastructure management

- Everything is code => Version controlled
- Envs indentical except for parameters => Guarantees
- Can tear down and rebuild infrastructure in < hour vs days or weeks
- DevOps is not the dark arts
- Started with our dev environment

  all the apps are up running
- Parts of prod networking are already

  place
- Just about certification now



Build Status Quo

- Status Quo is using Opsworks + Chef for builds
- Pros:
  - Helpful UI
  - Key management
- Cons:
  - SLOW BUILDS
  - Opsworks Chef runs have killed our boxen
  - Non-standard everthing
  - Outdated versions of Node
  - Random deploy fails

Better builds

- Adding tests for our Chef scripts using Test Kitchen
- Want to move to Packer
  - Cooks a VM and stores it as an AMI
  - Deploys happen much more quickly
  - Guarantee of matching environments
- We’re still figuring this part out

More QA Automation

- Started writing integration tests
- No point in adding automation until you have the tests
- Take 15 minutes every day an write an integration test
- We have set up a Jenkins server. Smoke tests -> 15 minutes from 8 hours
- Still figuring this part out too
- Getting to continuous integration requires a lot of investment, move towards it piece by piece

Questions?

Demo

- Chef
- Terraform
- Packer
- Node
- Redis
- NGINX
- https://github.com/dmfenton/featureservice-replicator

Questions?

@dmfenton

Further Research

- http://nathenharvey.com/blog/2014/05/01/devops-explained/
  https://sethvargo.com/the-ten-myths-of-devops
  https://www.youtube.com/watch?v=_U3CUBwb1lk
  https://www.youtube.com/watch?v=oX8af9kLhlk
- **https://www.chef.io/chef/**
- **https://www.packer.io/**
- **https://www.terraform.io/**
- **https://jenkins-ci.org/**
- **https://travis-ci.org/**
- **https://gitub.com/dmfenton/featureservice-replicator**

esri®