GOTO CHICAGO 2017

Reining in Chaos

Colleen Stock, Uptake Engineering


colleen.stock@uptake.com


@uptake

UPTAKE™

OUR MISSION

Uptake will be the actionable insights platform that makes the world more productive, safe, and reliable.

How Uptake Grew

- Uptake is 33 months old (2.75 years!)
- There are roughly 700 employed
- Hire approximately 25 people a week
- Currently about 400 developers
- Started with one vertical, expanded to 7

What Was So Broken?

- Uptake's deployment process (or lack of process)
- From 10 devs to a 200+ with no change in the process
- From 2 applications to 100+
- Without a unified deployment process, there was no way to keep track of what got deployed when and where.

How Does This Apply To Me?

- Could your team's deployment process use some work?
- If not, what about some other process in place that isn't working?
  - Tickets taking forever to get closed
  - Code review by people who don't know what they're reviewing
  - Weekly meetings that serve no purpose
- Ways to manage chaos and where to start

The Beginning

- We decided on Marathon/Mesos — Why?
  - Rest API
  - Easy app configuration
  - Docker artifact support
  - Load-balanced deployments
- Environment specific config
- No service discovery (which is why you need an environment specific configuration)

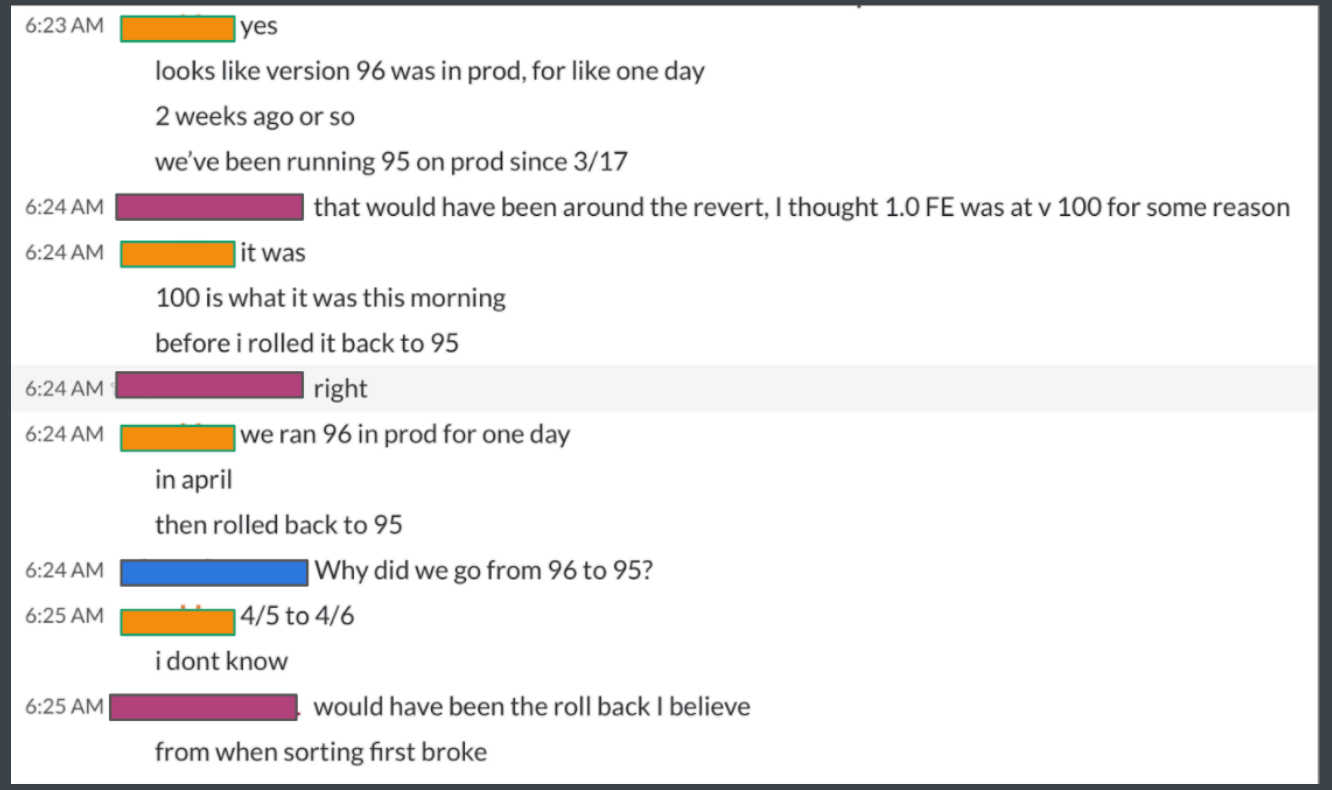
The Mess

- Application configuration: Mixture of Spring properties files and environment variables
- Marathon JSON: Template + sed script with environment variables defined in Jenkins jobs
    - Worked okay for 2 apps and 3 environments = 6 jobs
    - Apps went from 2 to 20 and the envs went from 3 to 6 = 120 jobs (Completely unmanageable!)
- No contract testing
- Regression testing against a specific set of app versions and configuration

Application Deployments

HOW THEY USED TO WORK:

- Jenkins jobs per application and environment
  - uptake-core-QA
  - uptake-core-RC
  - uptake-core-STAGING
  - ….
- Deploy apps one at a time to an environment



How To Recognize Chaos

WHAT INDICATED THAT THERE WAS A PROBLEM?

- No change in how things were done
- Confusion around what was being deployed
- Lots of time lost to putting out fires
- Deployments started to take forever
- Human-managed process, making it error-prone

How To Communicate The Problem

- Talk to people about it
  - Devs who have to deal with it
  - Managers who don’t know why things aren’t getting shipped out faster
  - Mention the obvious
- Diagram the current structure to show how ugly it is
  - But have a proposed solution in place

The Plan

- Synthetic YAML processing (Environment, group, and application level configs)
- All configuration and versions are promoted in a monolithic manifest
- Resolve the issues of having hundreds of configurations dispersed and configured differently and hard to parse
- Auditing and approval tracked as git PRs

How Did We Fix It Without Breaking Everybody Else's Stuff?

- Single Jenkins job per environment (All apps are in there)
- Hard work of gathering all the configs was done by my team
- Wrote small ruby app to do this
- Fully owned by the team (with stakeholder by-in)
- Ability to deploy sets of applications to an environment based on what changed (No longer human-managed!)

How To Fix Chaos

- How to communicate change
  - Nemawashi
  - Multiple modes
  - Education!
- Distributing pain
- Sunset the old process/application visibly

The Next Step

- We've only covered the first step!
- 170+ applications
- Modularized deployments
- Releasing customer-specific code separate from the overall Uptake Platform
- Release schedules differ across customers

Thank you.

Colleen Stock, Uptake Engineering


colleen.stock@uptake.com


@uptake

UPTAKE™