Hyrax in Production

Lessons from the First Year

Mark Bussey - Data Curation Experts

DCE

Developing Digital Repository Solutions

My friend River

April 2018
The bad old days

- Unexpected outages
- Difficult to diagnose
- After-hours maintenance
- Planned work disrupted & delayed
- Bess barricaded in her office

September 2018

How we live now

- Rigorous dress rehearsals
- Low-risk deployments
- Simplified configuration
- Self-healing systems
- Planned time with family instead of unplanned time working

What changed?

Disclaimer

Think about what you're doing holistically, and pick a few tools that cover your entire problem domain and solve all of your problems with them.

Dan McKinley
Choose Boring Technology

Perilous PDFs

- Klaxon/Andon - monitoring you'll react to
- HoneyBadger - wait, there are no errors here...
- Log Aggregation (Splunk) - what's in the log (which log)
- Self healing - gives time to diagnose the cause instead of reacting to the symptom
- Dress rehersal - test it, script it, commit it, re-deploy it, re-test it,
  risk to prod = low
  confidence in solution = high

Box backgrounding

- Will this ever succeed without manual intervention?
  - Yes = implement self healing
  - No = dig for root causes
- When Box's time dependent access keys are expired, it's API returns html 200 Success:
  <body>we couldn't attach your file</body>
- Solution: Add better error detection and separate Box processing in its own high-priority queue

Migration

Legacy Content + Hyrax 1.x→2.x

- Dress rehearsal
  local → CI → qa → staging → production
  Repeat
- Reproducible builds via ansible
- Exact duplicates of production environment via cloning
- Read only mode
- Temporary scaling via virtualization

Code and infrastructure are plentiful,

your attention isn’t

DCE Tools

- Amazon Web Services
- GitHub
- Capistrano
- Ansible
- nagios
- Pingdom
- Honeybadger
- Splunk
- Katalon Studio (evaluating)

Practices

- Test with real data & capture edge cases when you discover them
- Introduce change in small increments -
  i.e. Continuous Integration
- Deploy early, deploy often -
  i.e. Continuous Deployment
- Shorten feedback loops
- Minimize complexity
- Swarm problems as a team

Concepts

- Infrastructure as code
- Self-healing processes
- A culture of production readiness
- Rich telemetry & monitoring
- Boring Technology
- Continuous Integration ≠Travis
- Automated Testing
- Playbooks & Recipes

Reading List

- The Phoenix Project,
  Kevin Behr, George Spafford, Gene Kim
- The DevOps Handbook,
  Gene, Kim, Jez Humble, John Willis, Patrick DeBois
- 2018 State of DevOps Report,
  Puppet + Splunk
- The Agile Samurai,
  Jonathan Rasmusson

Thank you