Why automating everything adds to your toil

Colin Thorne, SRE, IBM Kubernetes Service
Cam McAllister, SRE, IBM Kubernetes Service
“Just automate it”
What is toil?

“Toil is the kind of work tied to running a production service that tends to be manual, repetitive, automatable, tactical, devoid of enduring value, and that scales linearly as a service grows”  [Vivek Rau, SRE Book]
What is toil?

- Gets in the way of making progress
- Repetitive manual tasks
  - Incidents, Tickets
  - Watching dashboards, producing spreadsheets
- The key is to reduce the amount of toil
- Project improvement work adds features or reduce future toil
What is automation?

• Avoid manual tasks by getting computers to do it for us

• Computers don’t get bored and love repetitive tasks

• Humans make mistakes

  • Chatbots for ops
  • Self healing
  • Provisioning/deploying
  • Self service
Why automation adds to toil

In the beginning … automation reduced toil

Reflections → Identified toil → New automation → Happy SRE
Why automation adds to toil
But then automation started to add to our pile of toil

“I tried to use it but the bot doesn’t work anymore”

“The automation stopped working … about a month ago”

“Oh yes, we changed the api and the automation hasn’t caught up”

Rot has set in
Automation rots over time

Just like any code, automation needs constant care and feeding.
Automation rots over time

Just like any code, automation needs constant care and feeding

• Dependencies change
Automation rots over time

Just like any code, automation needs constant care and feeding

- Dependencies change
- Requirements change
Automation rots over time

Just like any code, automation needs constant care and feeding

• Dependencies change
• Requirements change
• SREs change
Automation rots over time

Just like any code, automation needs constant care and feeding

• Dependencies change
• Requirements change
• SREs change
• Production systems change
Automation rots over time

Just like any code, automation needs constant care and feeding

• Dependencies change
• Requirements change
• SREs change
• Production systems change
• Languages change (Python2 -> Python 3 anyone?)
Unused automation

• Automation written once by one team, but no one uses it

• Not publicized

• So we’ve spent effort to create it, but no one (or very few) people use it

• Yet we still maintain it
Duplicate automation

• How many different bots can you have to do the same thing?

• Not invented here leads to duplicate automation

“Of course that other team’s automation is good, but it doesn’t quite fit what I need. I’ll write my own”
Adds to system complexity

• How much can SREs keep in their brain?

• When automation doesn’t work, where to start looking?

“Ironically, although intended to relieve SREs of work, automation adds to systems’ complexity and can easily make that work even more difficult” [Seeking SRE, John Allspaw and Richard Cook]
Too many tools

• The more tools you have, the more you have to maintain

• Some tools may not get used for weeks or months

• Danger is that when you come to use a tool it doesn’t work
  • more wasted time either fixing or falling back to the manual way

• Not always obvious which tool should be used to solve a problem
Automation is Good!
Automation is Good!

Toil is Bad
Automation is Good!

Toil is Bad

Reduce the toil caused by automation
Minimising Rot Potential
Build as a Developer

Automation is development, so treat it as such

- Designed
- Clean architecture
- Properly tested
- Full managed lifecycle

- Deployed to a production system (not just running on my laptop)

- Properly maintained (yuck.. Toil)
Build Self-Service tools

• We don’t need Full autonomy

• Create self-service tools that anyone can use
  • people AND services (microservice architecture)

• Call other self-service tools provided by developers
  • Tied to product function & new changes – preventing rot
Igor

• Faithful assistant (see Discworld / Pratchett)

• Self service bot
  • Used by SRE and Developers over Slack

• Doer not a thinker

Igor bot (SRE) 10:34 AM
Lightning master! we need more lightning!
"If all you have is a hammer, everything looks like a nail"

• We want our tools to be used a lot
  • Design our hammer so that many errors behave like nails.
How big a hammer?

- Restart the container
- Reboot the machine
- Reload the operating system and reinstall it
- Re-provision the machine
- ?
- ?
- ?
- Delete the universe and recreate it
Oh look.. a nail!

Cameron McAllister (IKS SRE) 5:39 PM
reload 10.171.78.70 outage:0s

Igorina bot (SRE) 5:39 PM
Raising a new change request.
Change request raised: CHG0376706 waiting for approval
I am starting the OS reload of prod-dal10-carrier2-worker-1035 (10.171.78.70) now.
Maximising Usage
To minimize Rot
Usage Prevents Rot

• Place it somewhere prominent
  • Eg in a slackbot
  • Don’t hide it

• Make it as easy to use as breathing
  • Minimise barrier to entry
Usage Prevents Rot

• Promote the Automation
  • Playbacks & Education
  • Runbooks should point to the automation first

• Success is people raising Issues on your automation
  (as long as you fix them)
Dealing with Rot
You cannot avoid it forever
Minimise Effort invested

• Have a very strict MVP to see if it gets used
• Don’t deal with all corner conditions
• Defer to SRE if something unexpected happens
Survival of the Fittest

• Don’t assume that the current approach is the best one

• Encourage Innovation
  • Can we do this better a different way?

• Compare effort to fix rot with benefit of the tool running
When Rot has set in...

... It’s often most humane to put it down.

Cameron McAllister (IKS SRE) 12:18 PM
quiesce TERMINATE

Igor bot (SRE) 12:18 PM
Prod: But i served you so well.. I hope my replacement displeases you less
Igorina

- **Self service bot**
  - Used by SRE and Developers over Slack
  - Doer not a thinker

- **Enhancements**
  - Used by other microservices over a REST API
    (for self healing)
  - Auditable – via change management tooling
  - Sharded
In Conclusion
Summary

• Automation Good

• Toil Bad

• Reduce toil produced by automation
  • Build as a developer
  • Maximise use of your automation
  • Treat your automation as evolutionary steps

Contacts

colin_thorne@uk.ibm.com  LinkedIn: linkedin.com/in/cjthorne  @ColinJThorne

C.McAllister@uk.ibm.com  LinkedIn: linkedin.com/in/cam-mcallister
Questions?