

ALL DAY DEVOPS

NOVEMBER 6, 2019

Dan Illson

**Fences and Gates:
Designing Ops for
DevOps**



Agenda

Who am I? And what do I do?
Herding vs Chasing
Policy and “Guardrails”
Closing the Loop(s)



Who am I? And What do I do?

- @dan_illson
- Cloud Advocate @ VMware Cloud Services
- Former networking Sales Engineer
- Originally specialized in telecom provider architectures
- Customer projects moved me toward software based infrastructure
- SDN lead me to cloud, which lead to Cloud Native and kubernetes
- Development and API use naturally followed a cloud native focus
- Merging development and deployment with experience in scale and process



Herd^{ing} vs Chasing





Look, it's Operations!

Here are our applications



Would you rather chase?



Or herd?



Aim for the minimum number of "herds"

Think boundaries (fences) rather than granular controls



Policy and "Guardrails"



Two types of “Policies”

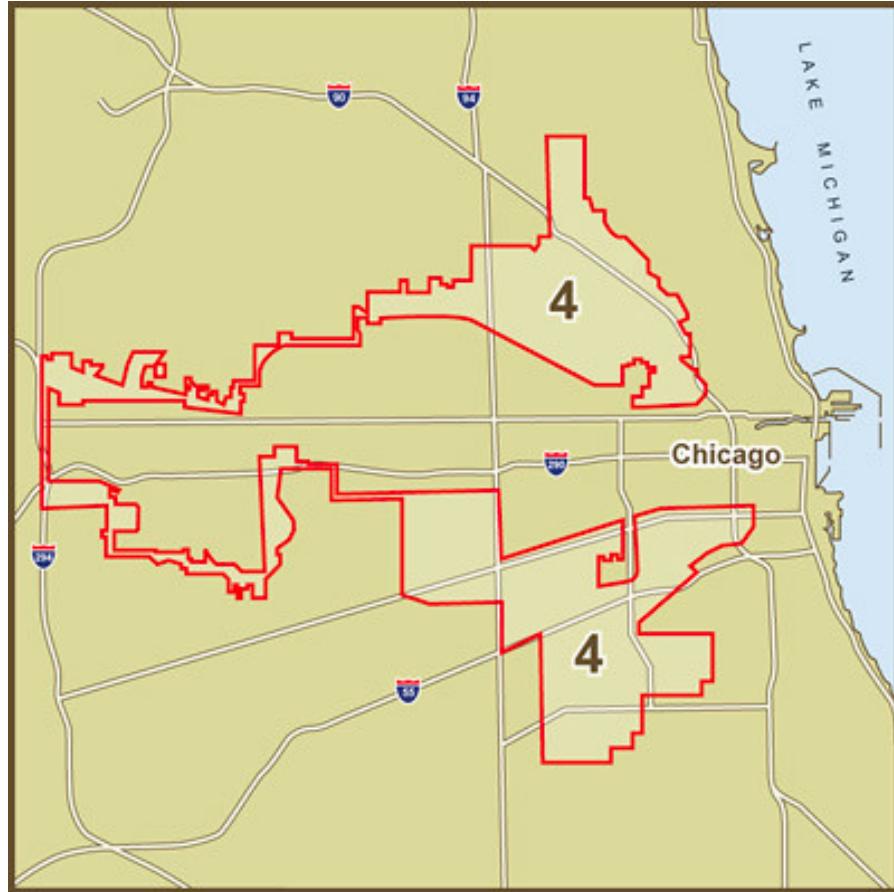
- “Big P”
 - Used by humans / teams
 - Correlation to business and organizational needs
 - Generally qualitative
 - “Deploy faster”, “Have fewer critical bugs”, etc
- ”Little P”
 - Easily interpreted / consumed by machines and systems
 - Alignment with technology and configuration
 - Constraints (quantitative)
 - “Object storage must be encrypted”, ”Port 22 not open to the internet”, etc



“Big P” Policies should be mapped into “little P” guardrails

Start general, only get as specific as you must

Specific -> General



General -> Specific

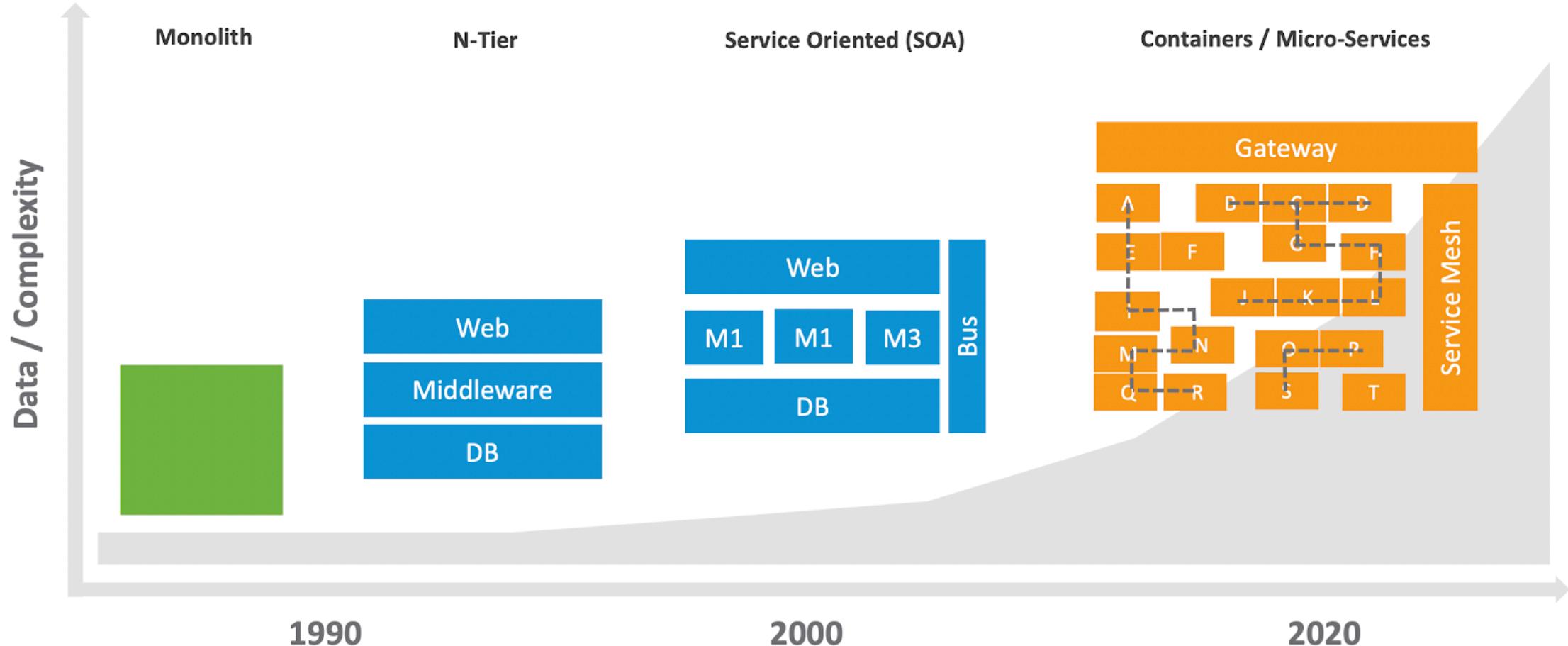


Store guardrails in version control
To track their evolution through time

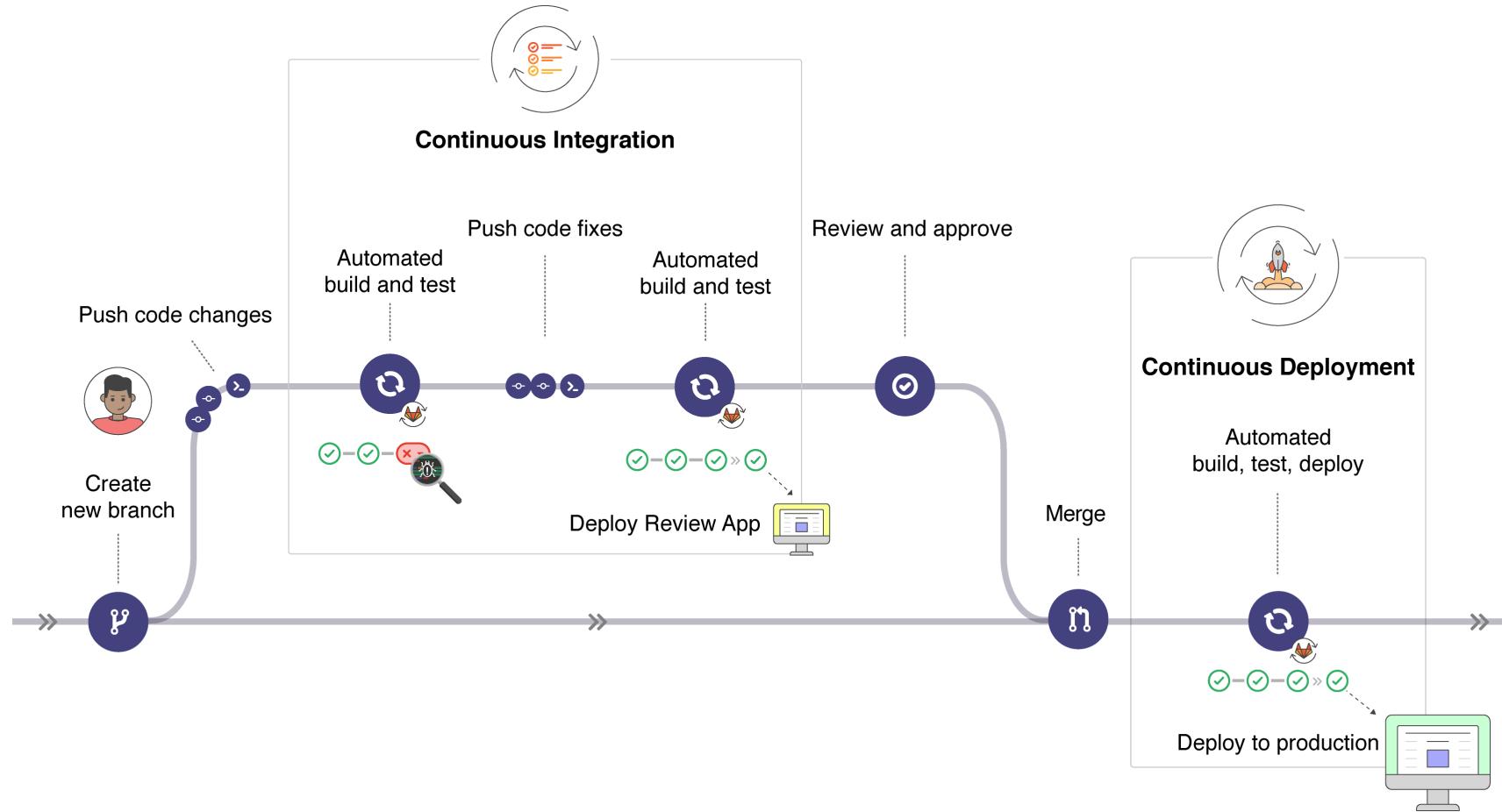
Closing the Loop(s)



Applications are more Complex ...



Pipelines Act on Declared Intent

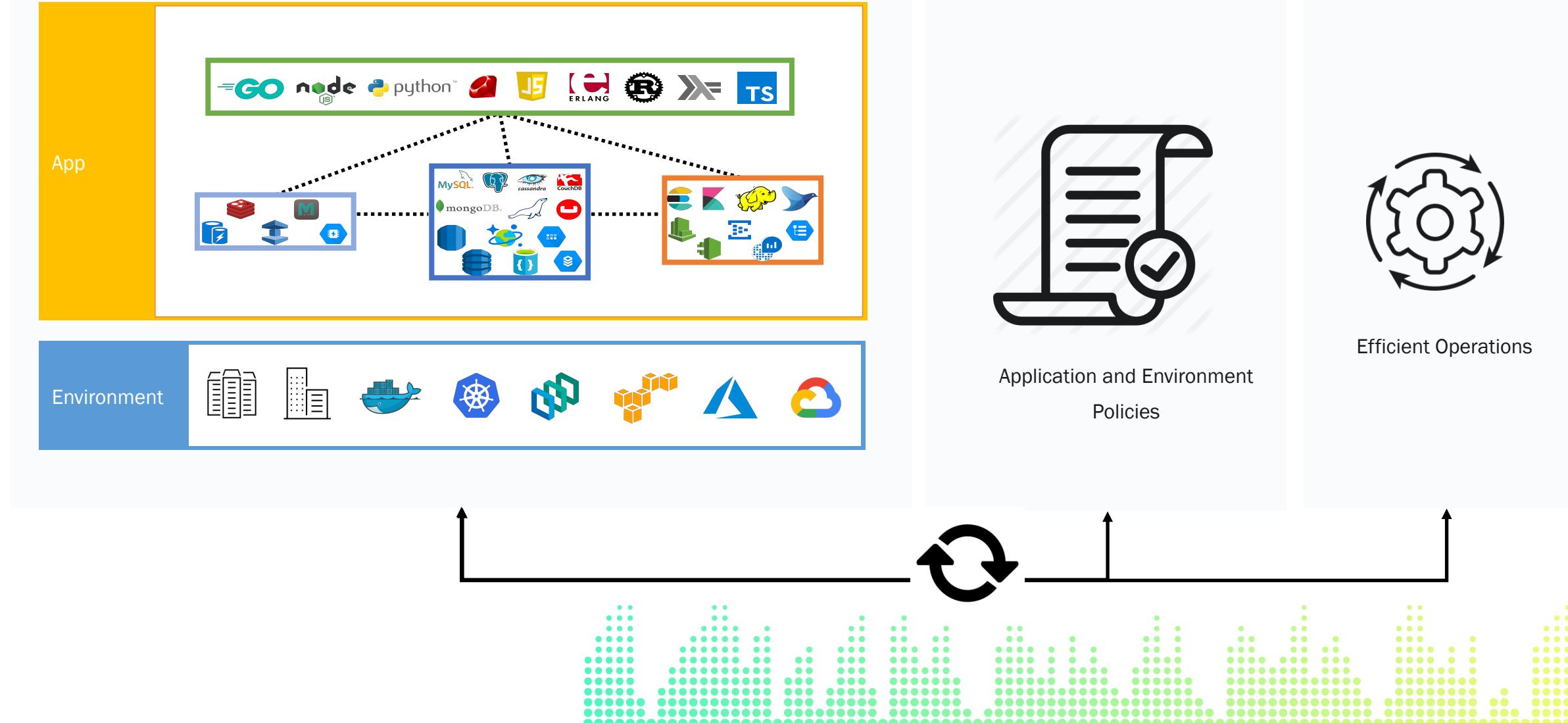


Guardrails are an intent

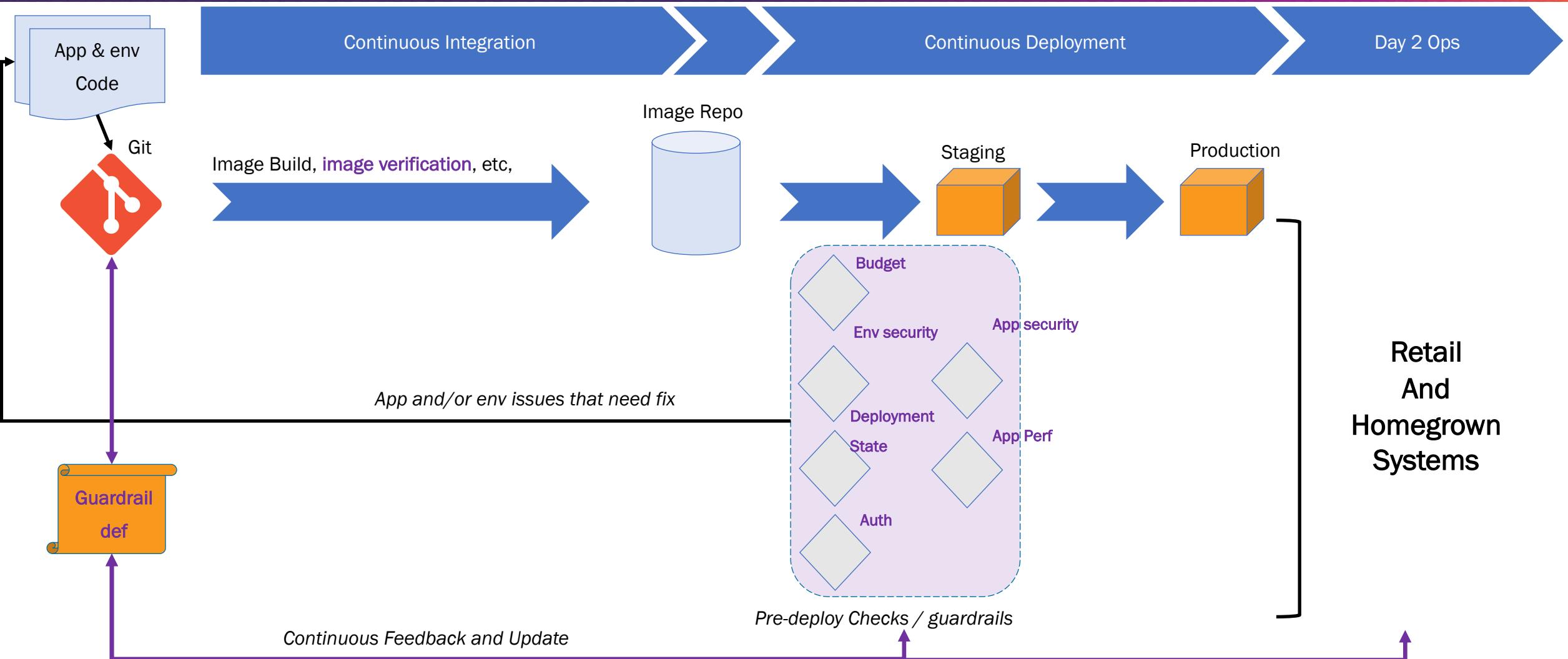
Should "Operations" be about the break/fix dynamic?

Or could it be about continually refining guardrails to avoid "break" events?

Ops in the Age of DevOps



Continuous Verification



<https://thenewstack.io/continuous-verification-the-missing-link-to-fully-automate-your-pipeline/>

Feedback loops are an optimization process

Operations can evolve with applications and policy, rather than oppose them

www.cloudjourney.io
[@cloudjourneyio](https://twitter.com/cloudjourneyio)



