

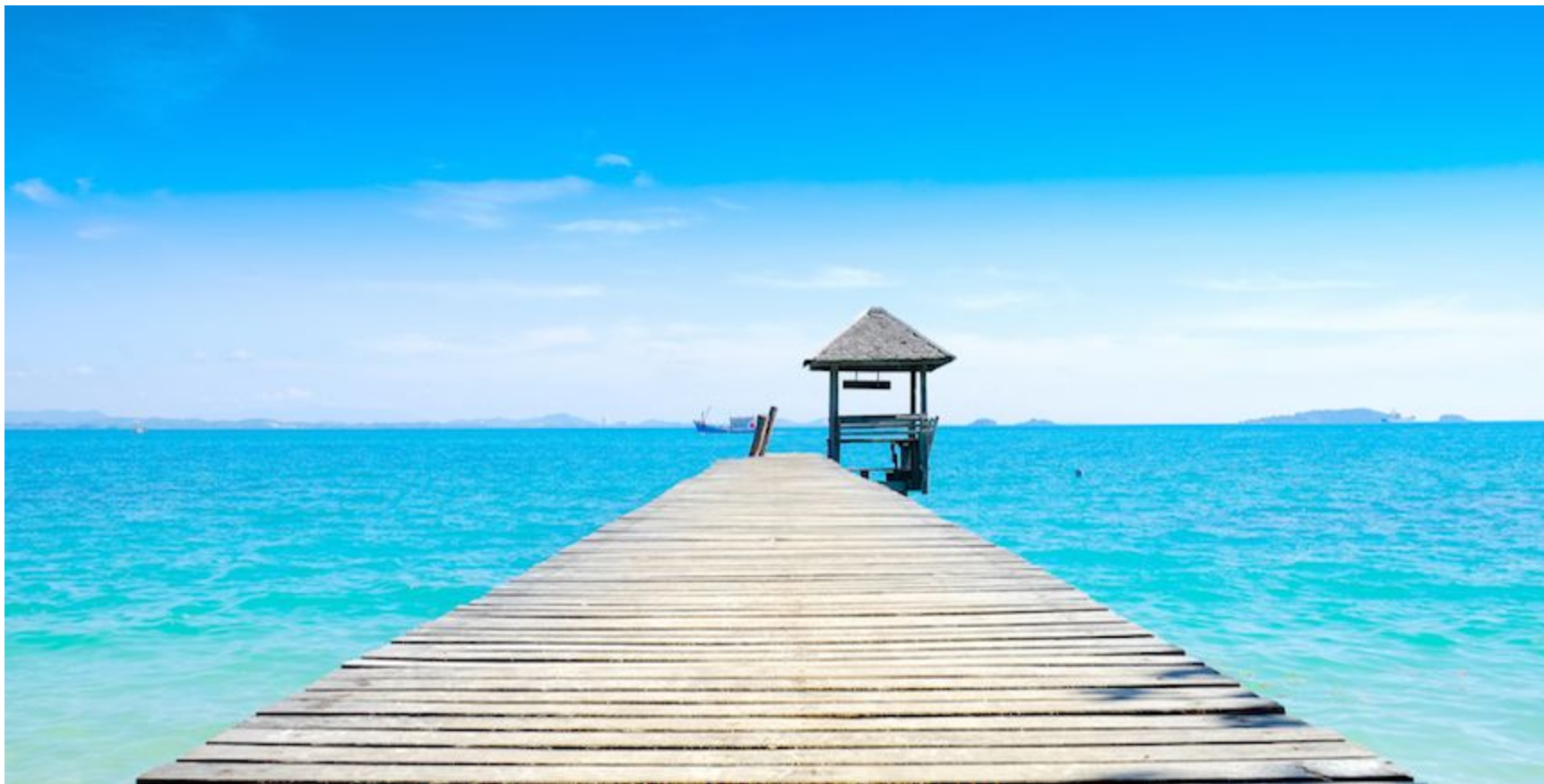
Cloud-Native Progressive Delivery

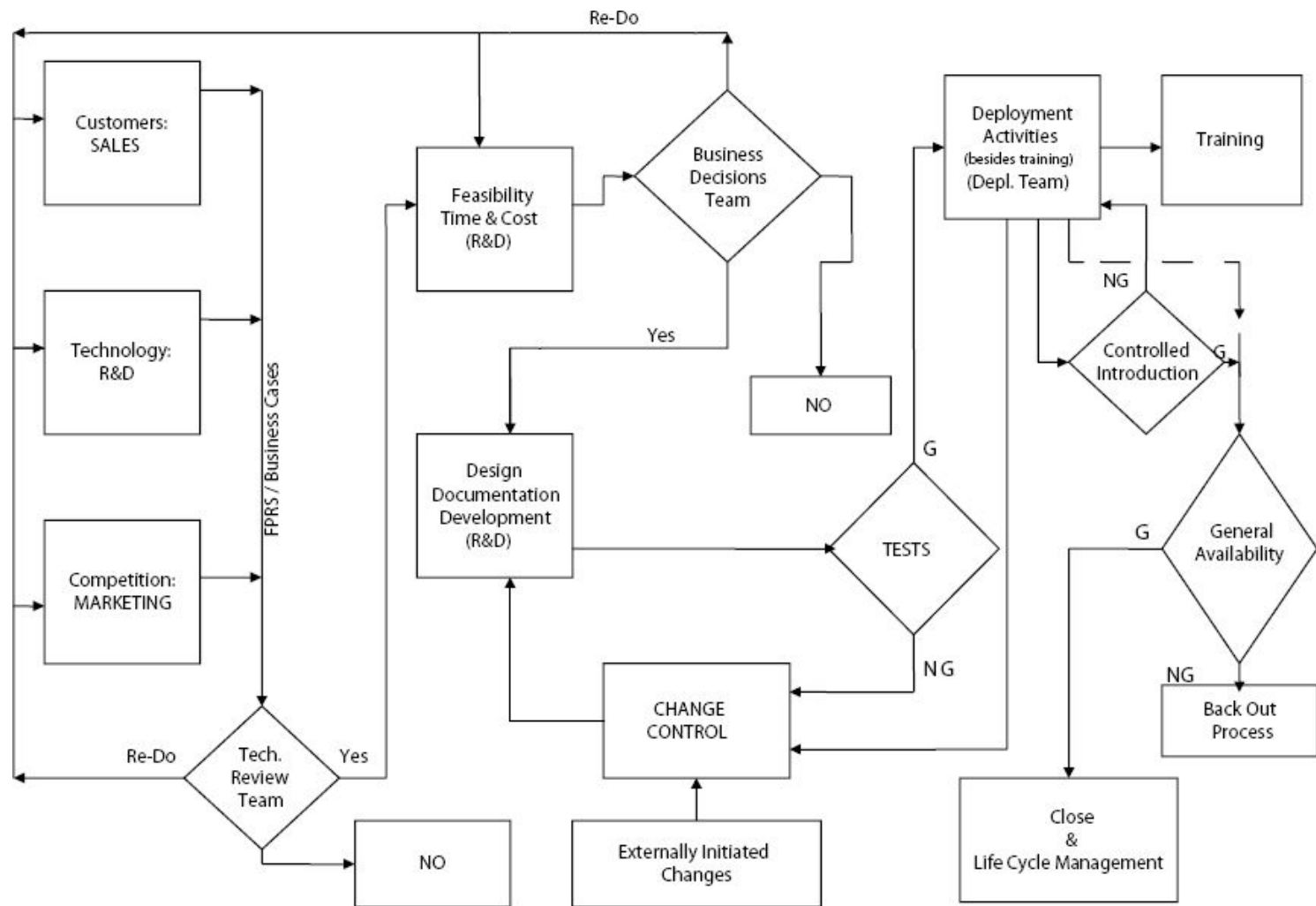
GOTO, Copenhagen Denmark | November 2021

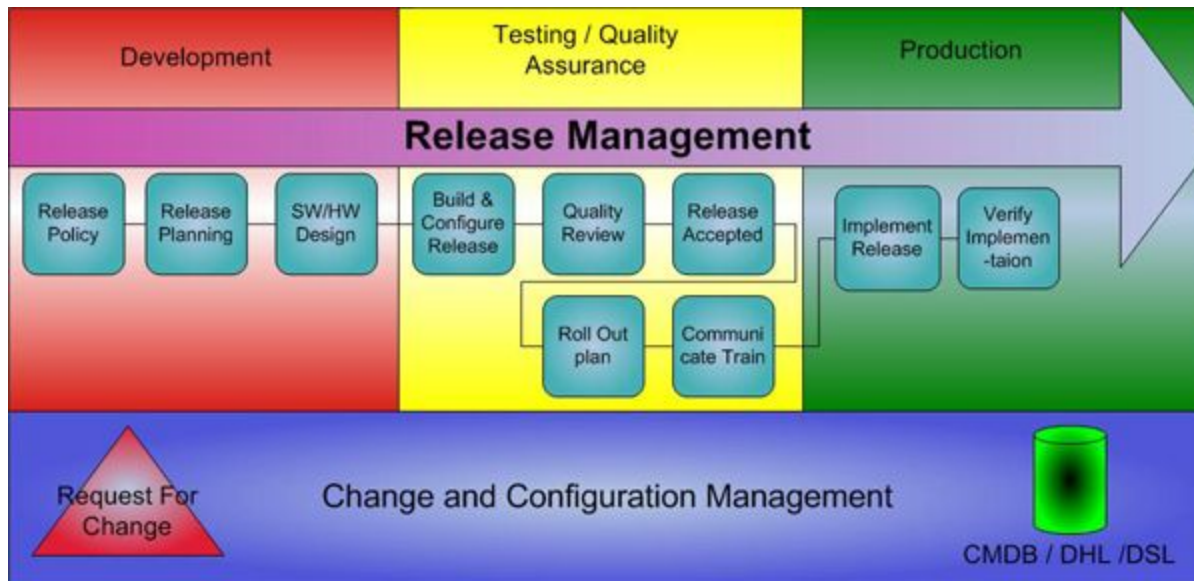


Matt Turner

@mt165 | mt165.co.uk



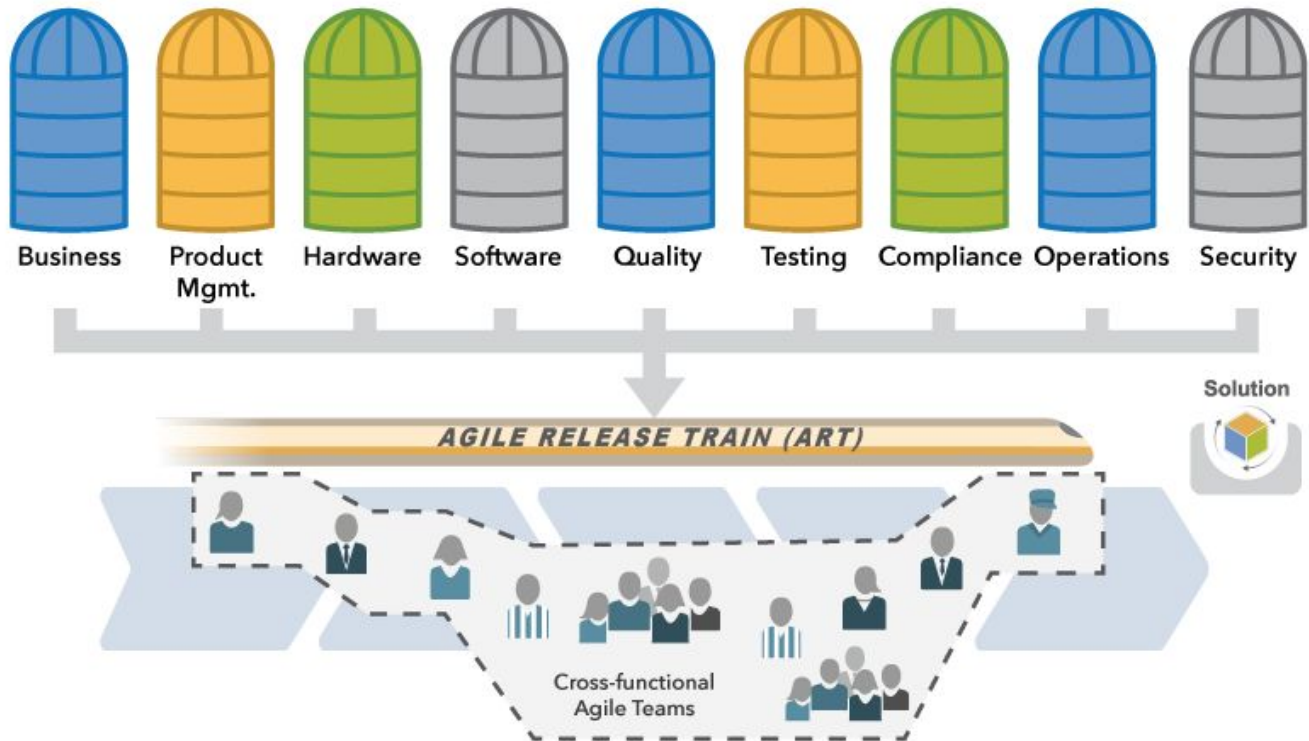




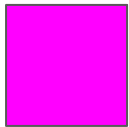
SOFTWARE DEVELOPMENT LIFE CYCLE

Phases, Models, Process and Methodologies

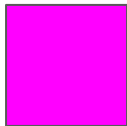
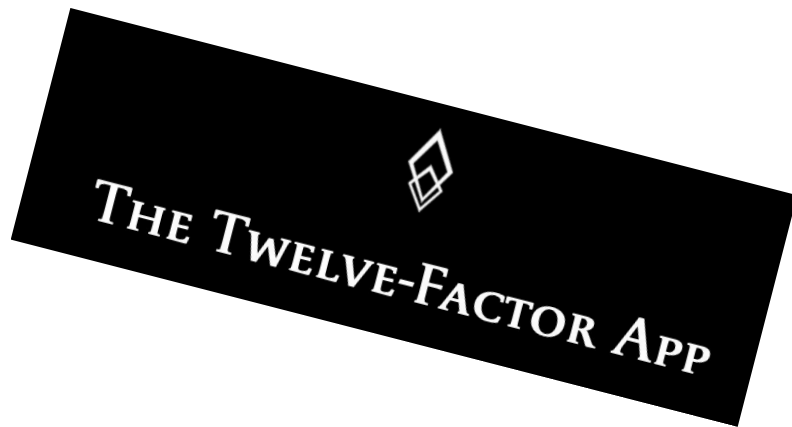




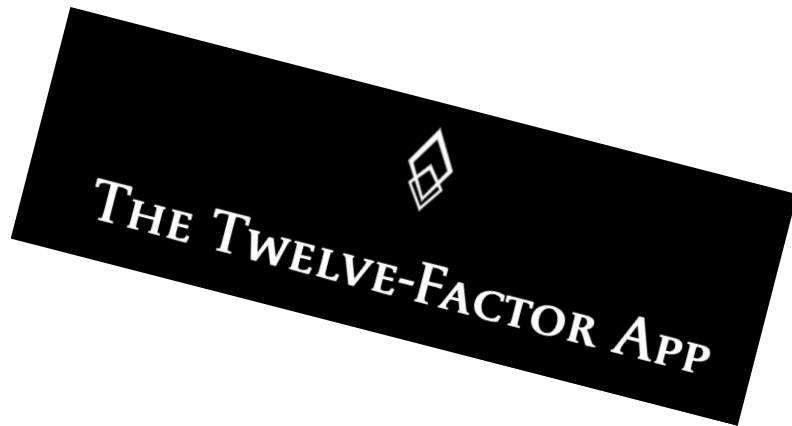
How Does Cloud-Native Enable This?



12 Factor Apps



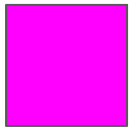
12 Factor Apps



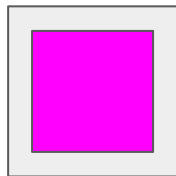
I. Code in git



III. External Config

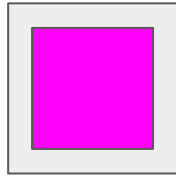


Docker



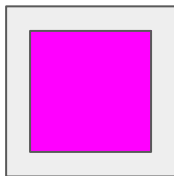
Docker

II. Isolate Dependencies

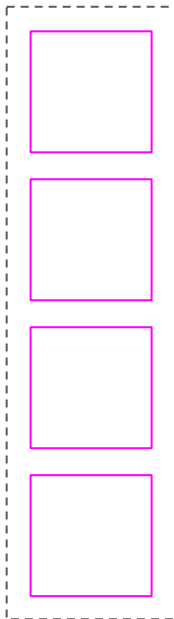


Docker

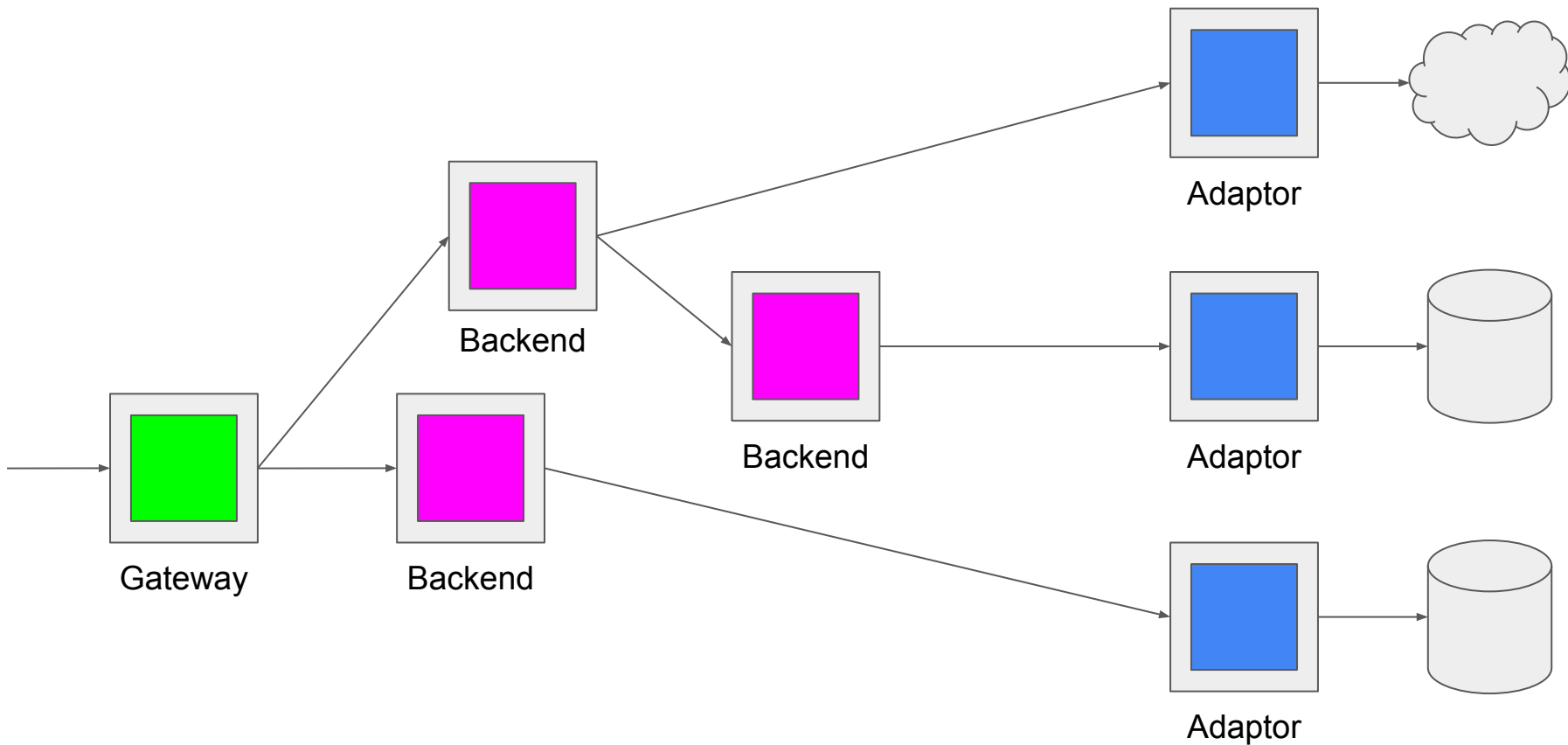
II. Isolate Dependencies



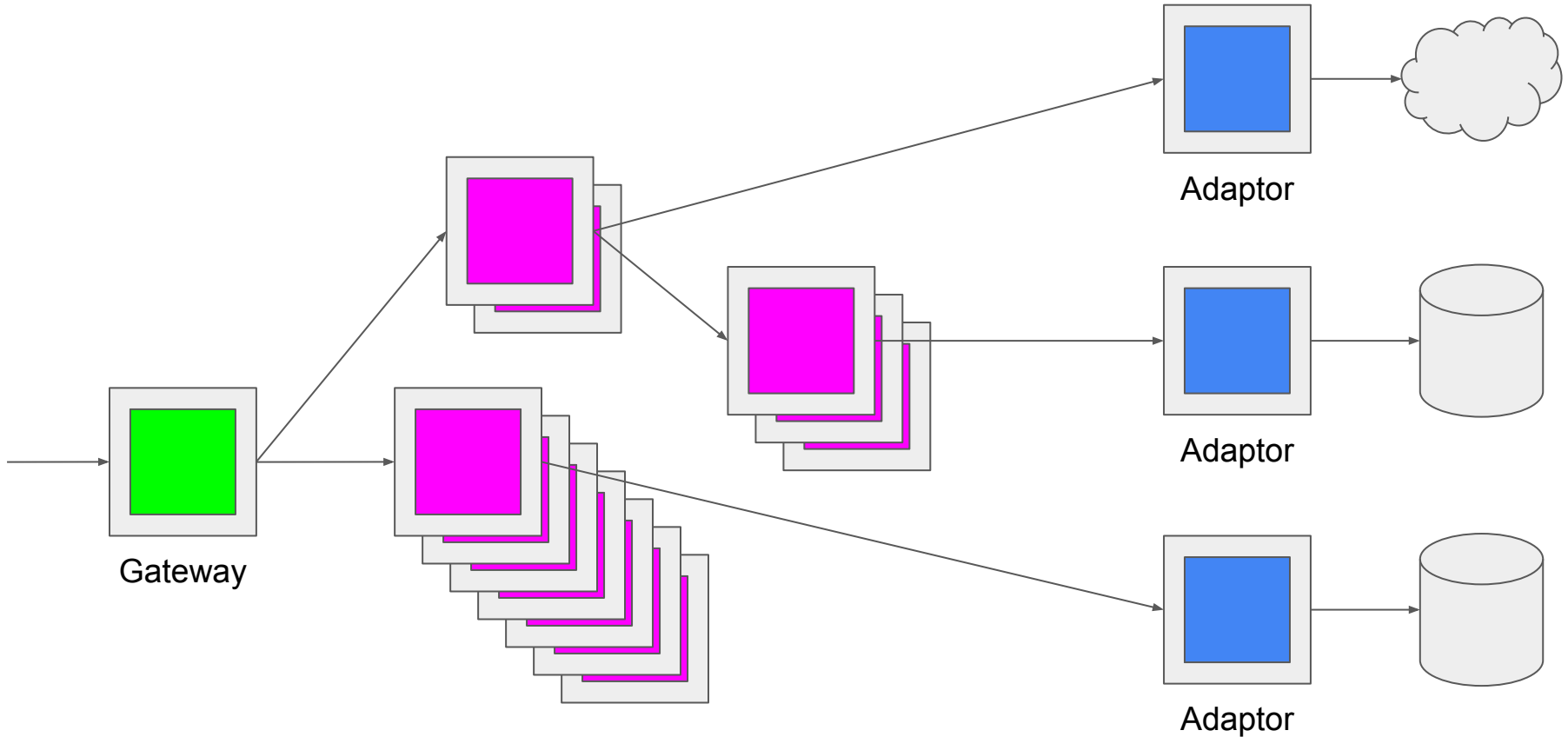
V. Strictly separate build and run



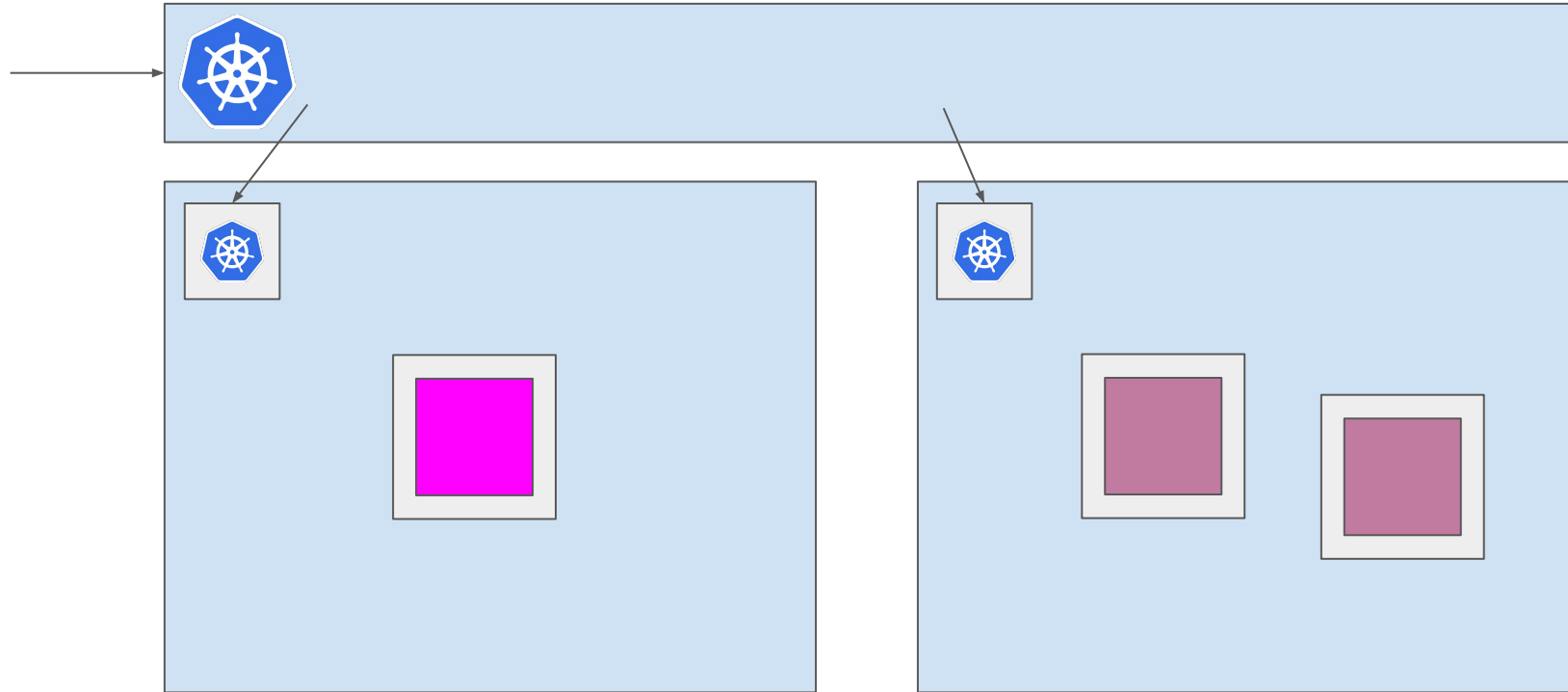
Microservices



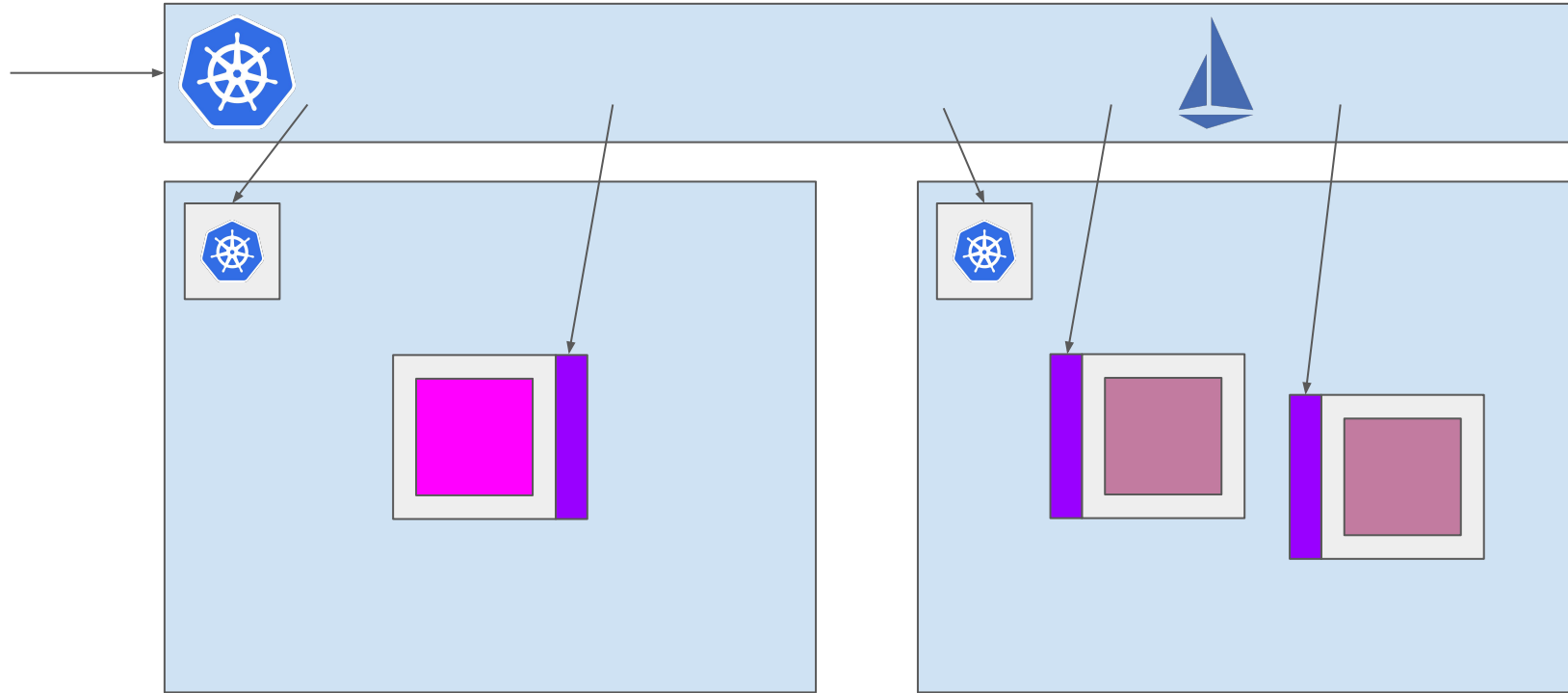
Microservices



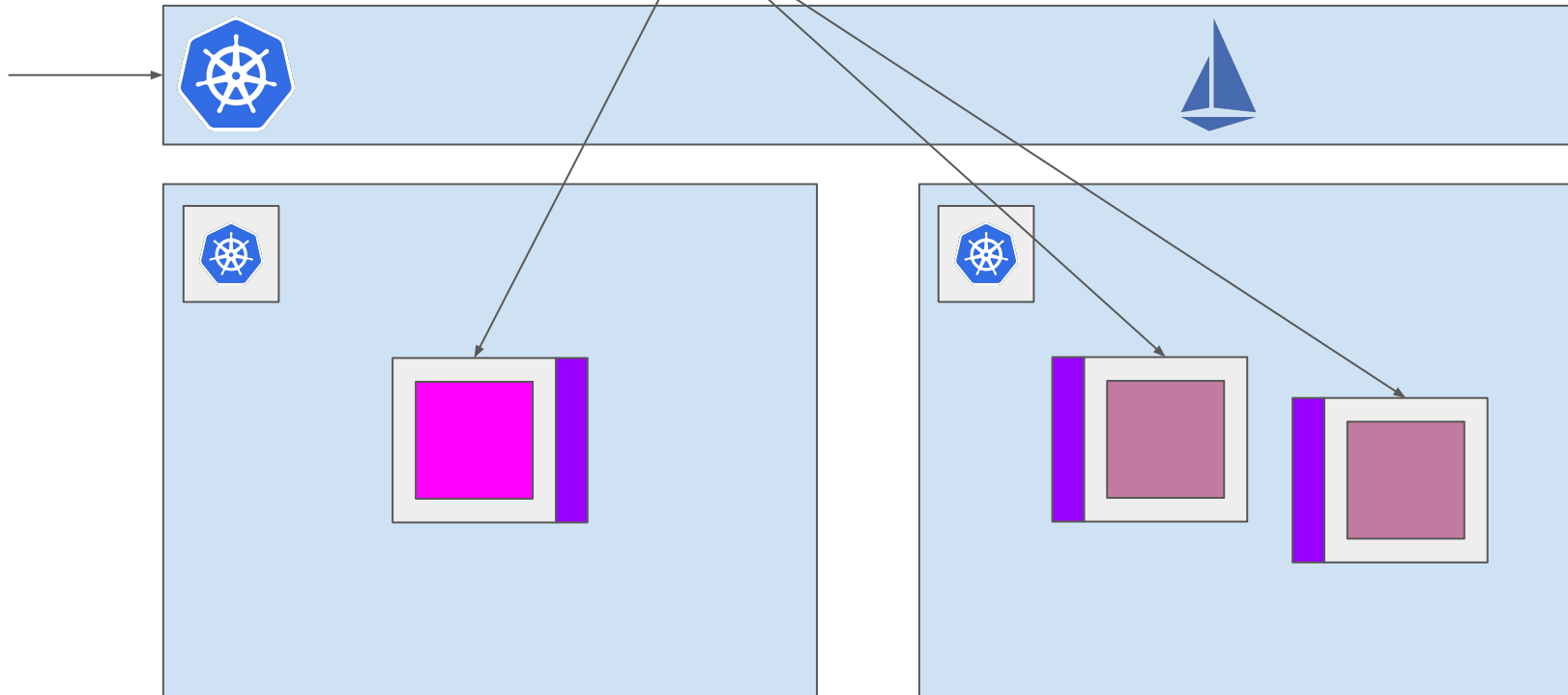
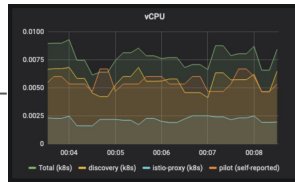
Kubernetes



Service Mesh



Metrics



Metrics

“RED”

- Rate - requests / second
- Errors - errors (%)
- Duration - latency of responses



Service Levels

- SLA - Service Level Agreement - broad statement of what's on offer, reads like a contract
- SLO - Service Level Objective - measurable, quantified target for availability, performance, etc. Eg error rate %, latency ms.
- SLI - Service Level Indicator - how will we measure the service level? How are we measuring things? Where? How are we aggregating them?

Infrastructure-as-Code & Declarative Systems

- Everything is described as “code” (needs an API)
- Eg Terraform, Kubernetes YAMLs
- No more click-ops!

GitOps

- Git as the single source of truth for everything
- Uses IaC to describe the *desired* state
- Committed to git
- Reconciled to the world
- Enables Operations via git (rather than by ticket)

Deployments Now

Continuous Integration

- These days actually means Continuous Build
- Original meaning still relevant and coming later

Deployment

- Taking a software package and running it

Continuous Deployment

- Deploying every time there's a new build

Progressive Delivery

Release

- Exposing a piece of software to *users*

Continuous Release

- Exposing users to every new Deployment
- => Exposing them to every new Build

Deployment ==
Release?

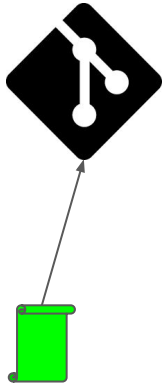
Deployment != Release!

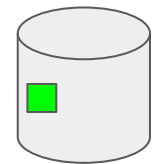
We Have the Technology!

X. Dev/Prod Parity

Keep development, staging, and production as similar as possible

Build

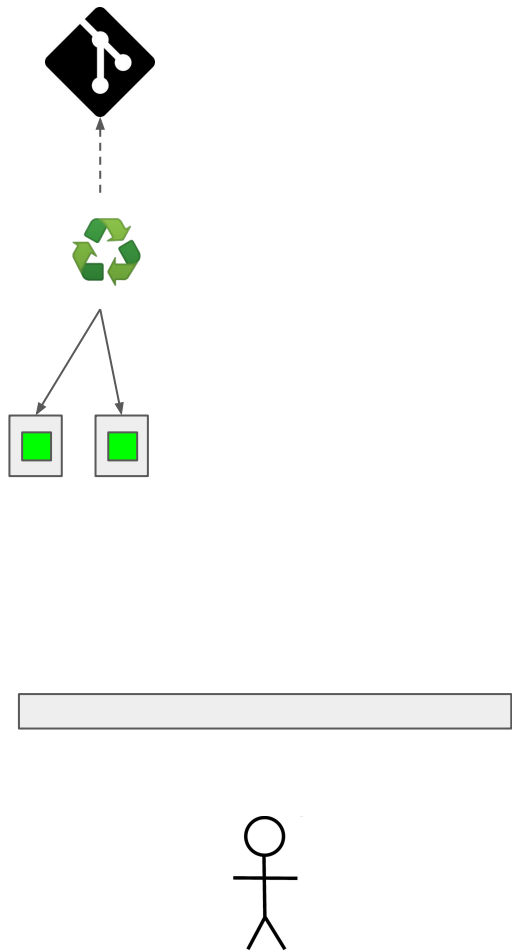
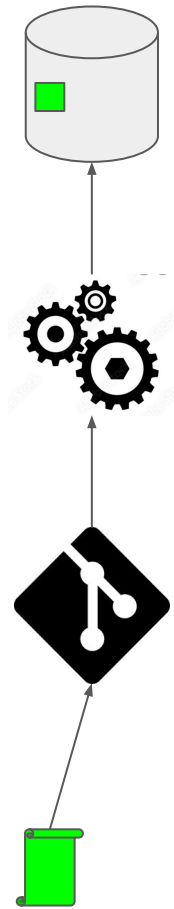


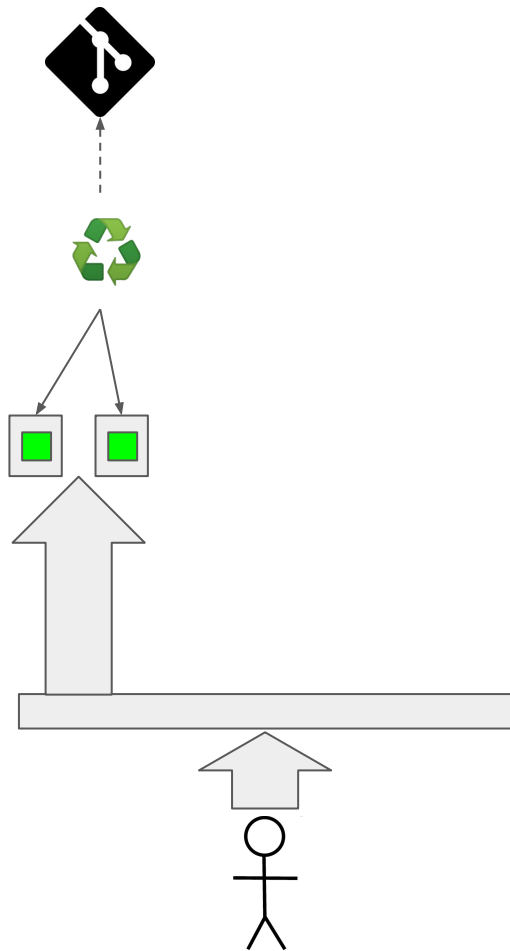
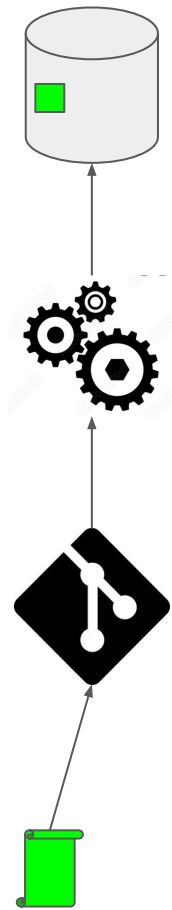


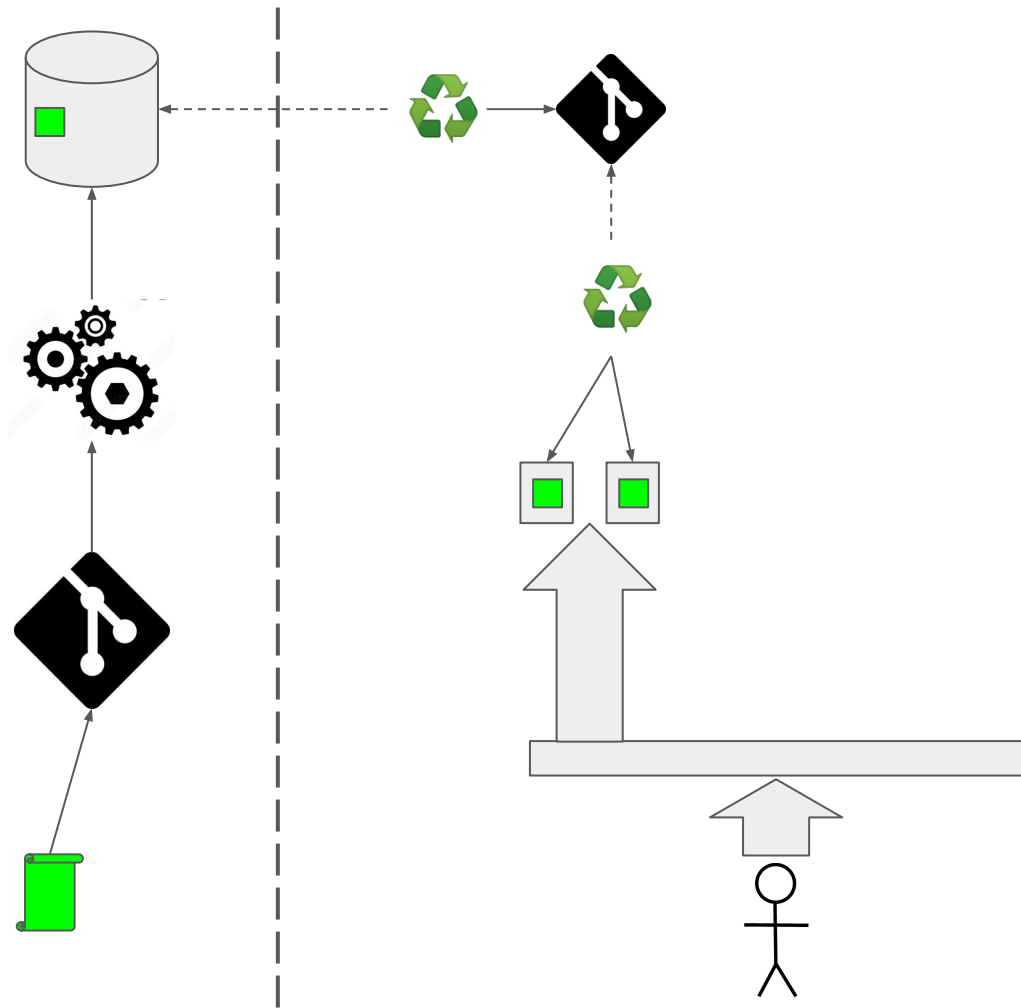
Contract

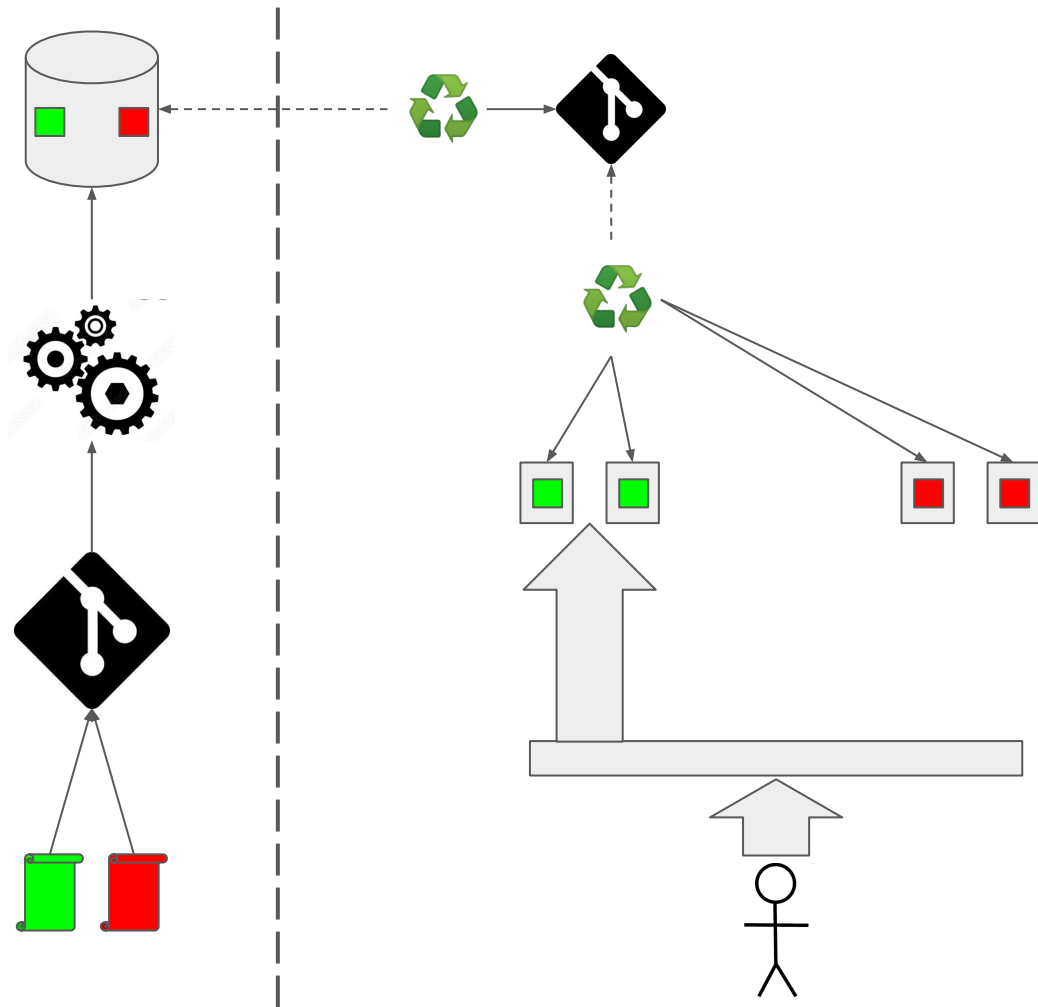
- Triggered by a new commit to main
- Produces a new container image and push to the registry
- Bottom of the testing pyramid: Linting, Compilation, Unit Testing

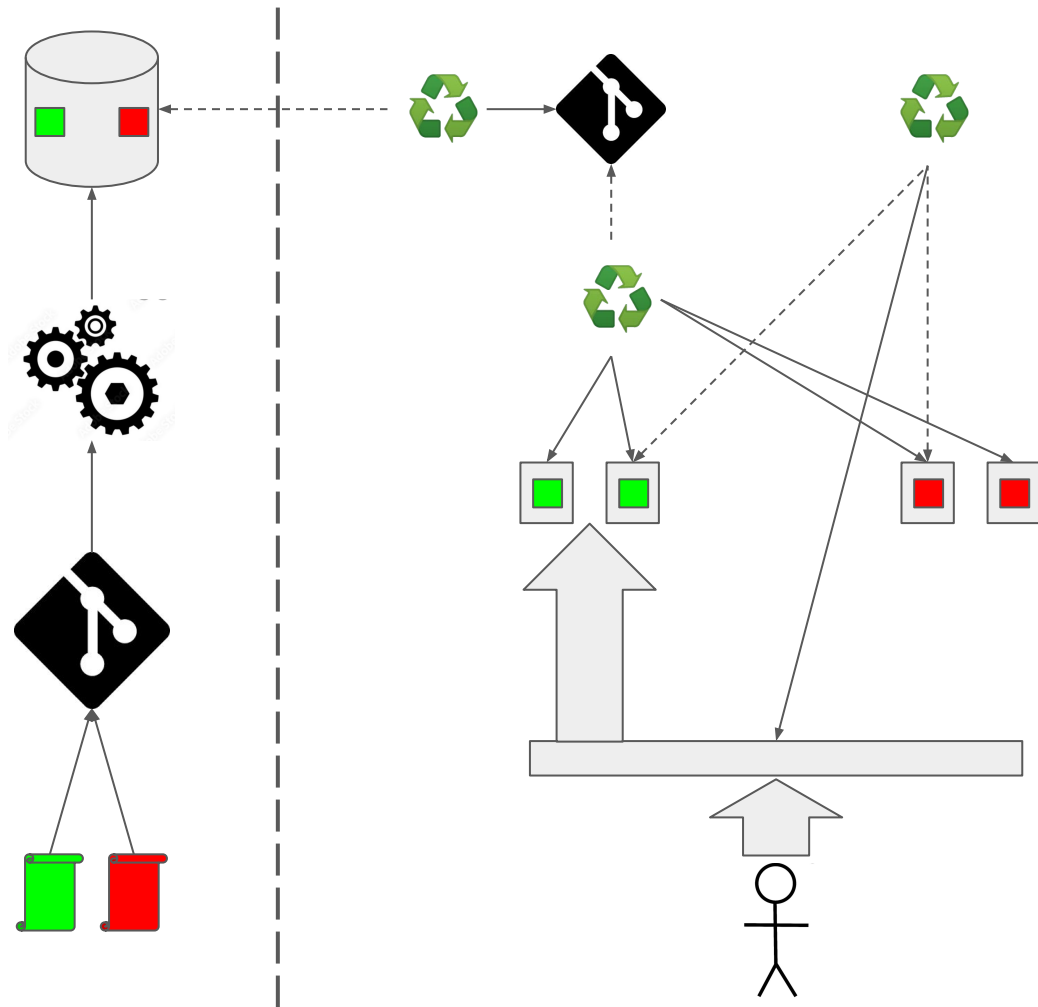
Deploy





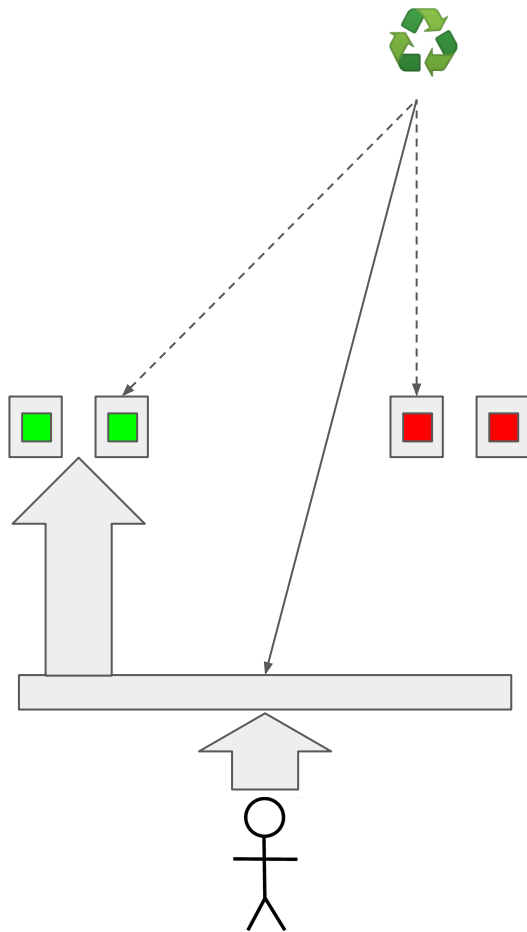
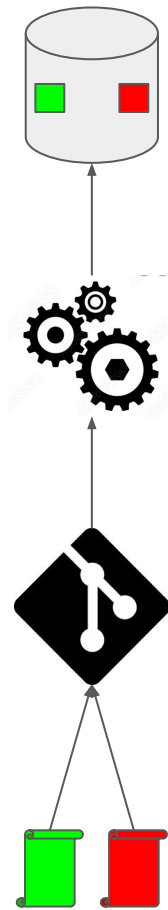


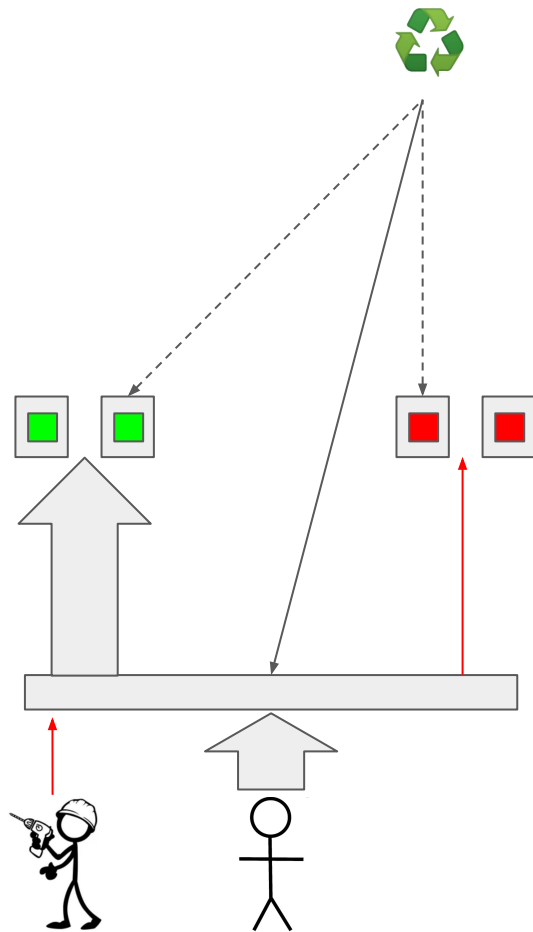
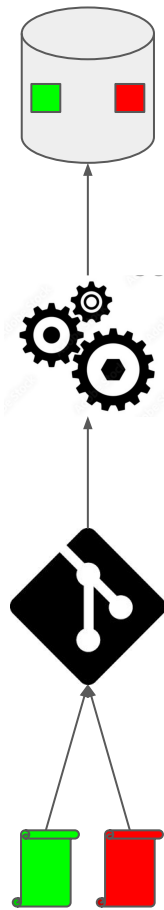


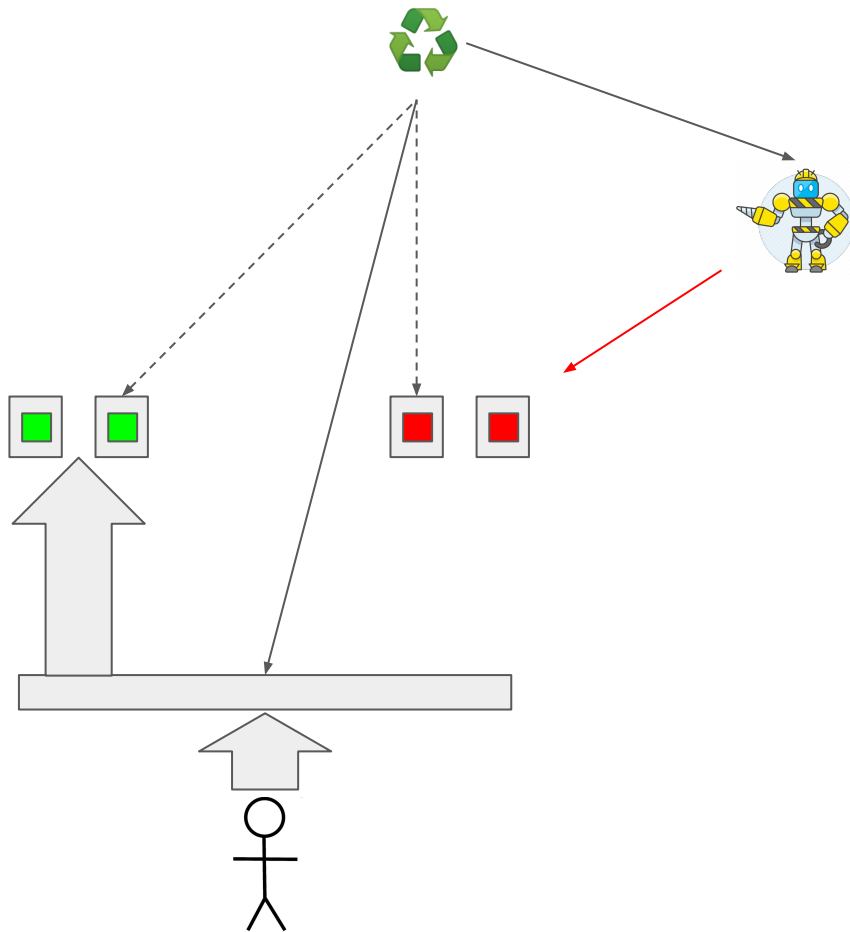
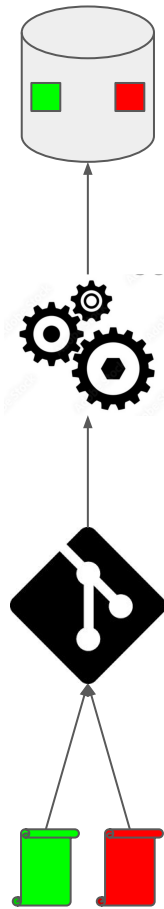


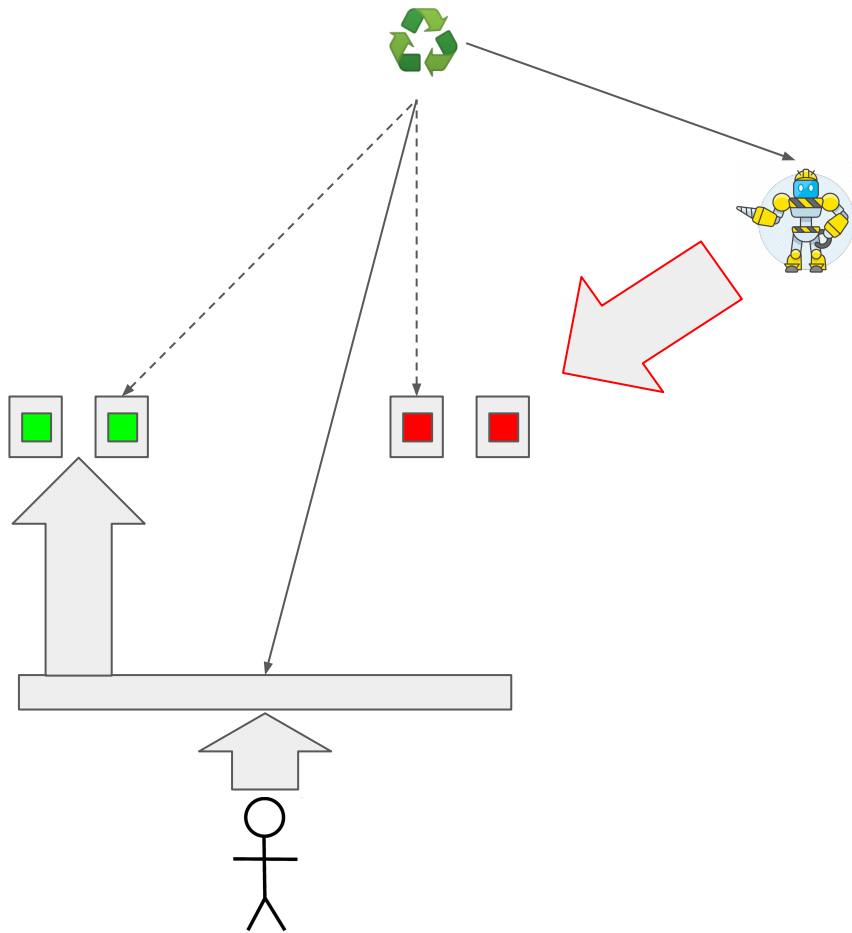
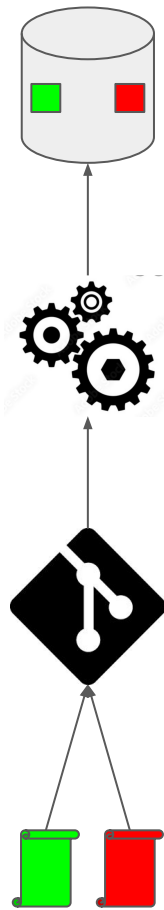
Contract

- Triggered by a new image appearing
- Deploys to prod cluster, prod namespace
- No user traffic



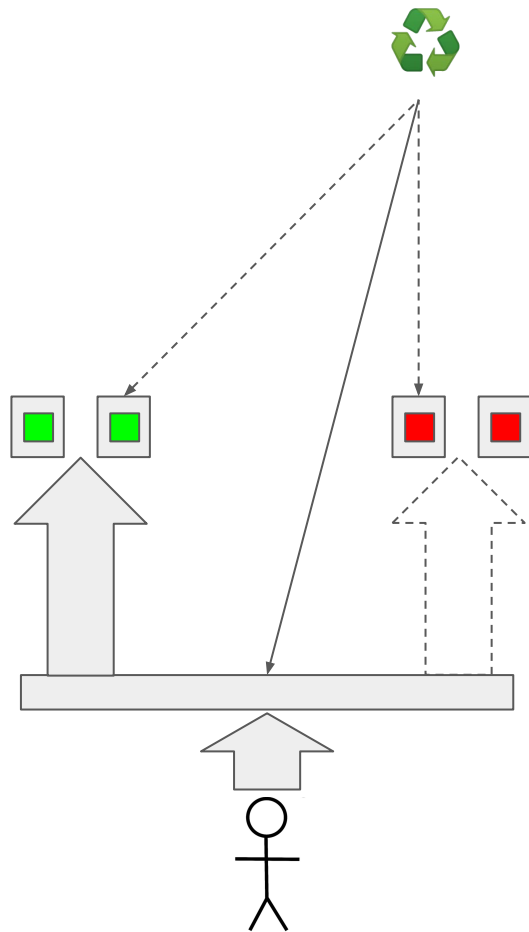
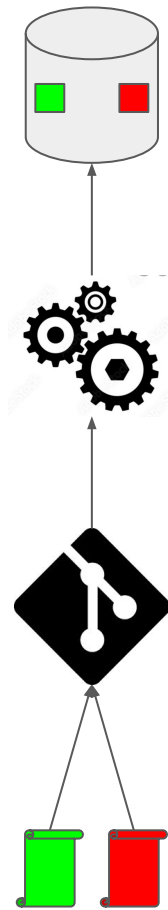


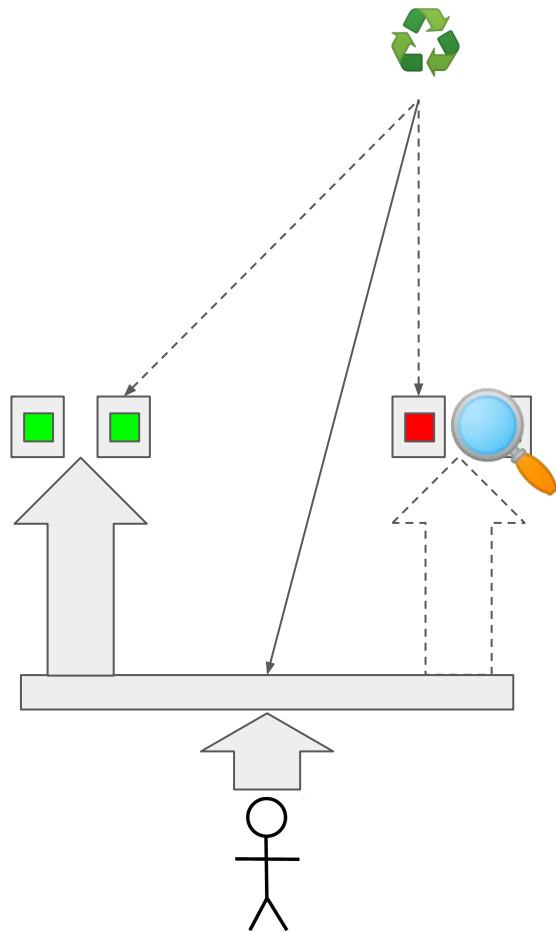
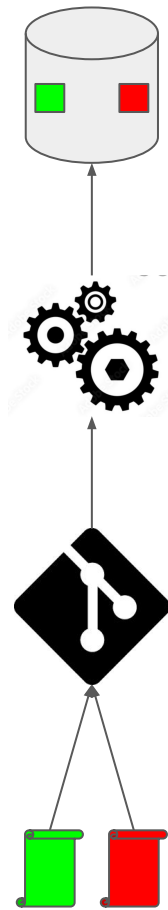


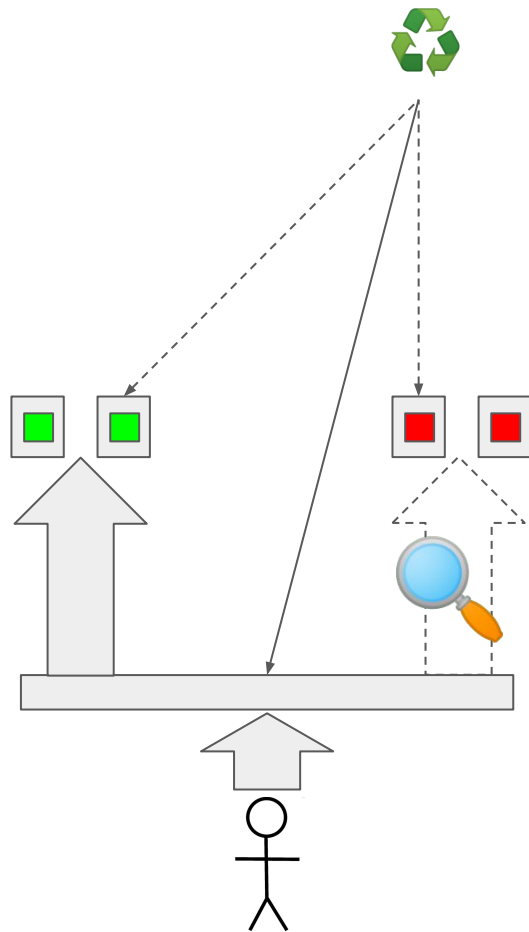
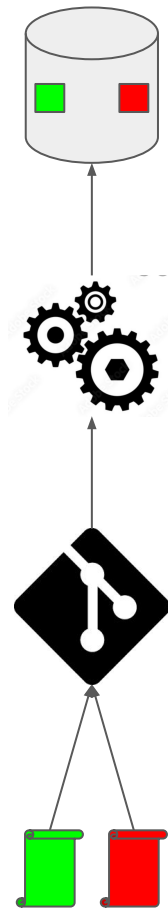


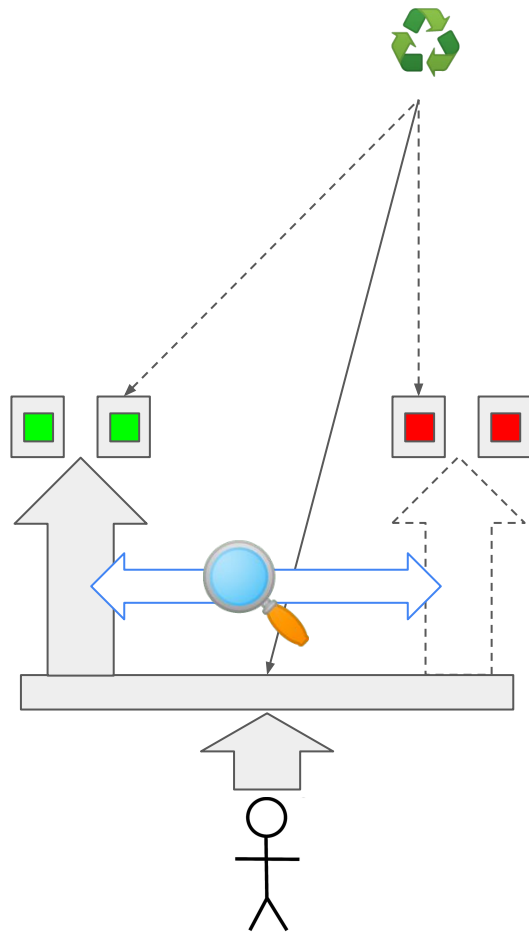
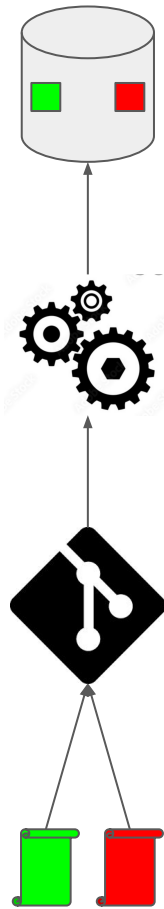
Isolation

- Does it even start?
- Available for manual testing
- Automated integration testing
- Automated end-to-end testing
- Automated non-functional testing
 - Failed if performance isn't within SLO





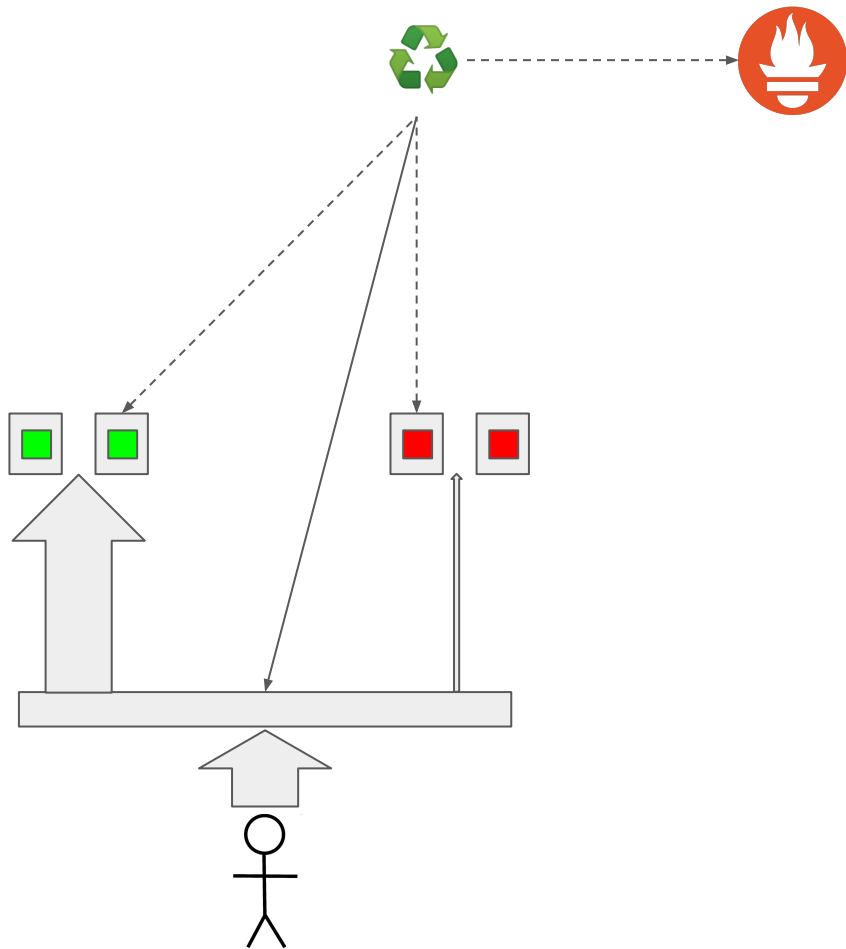
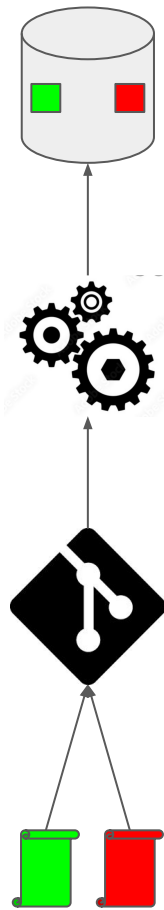


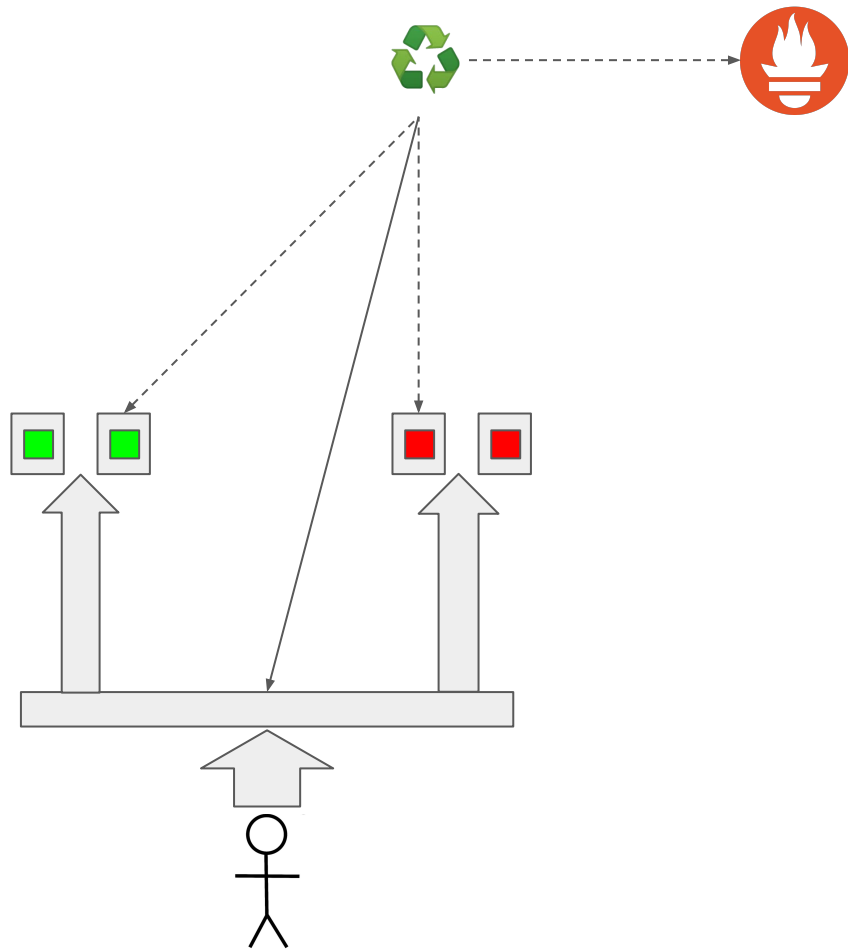
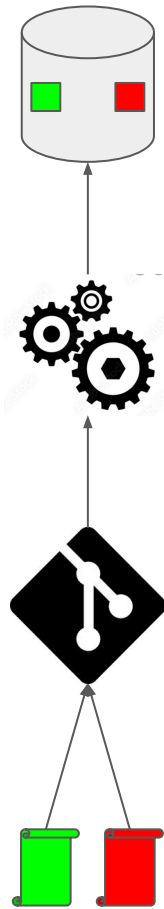


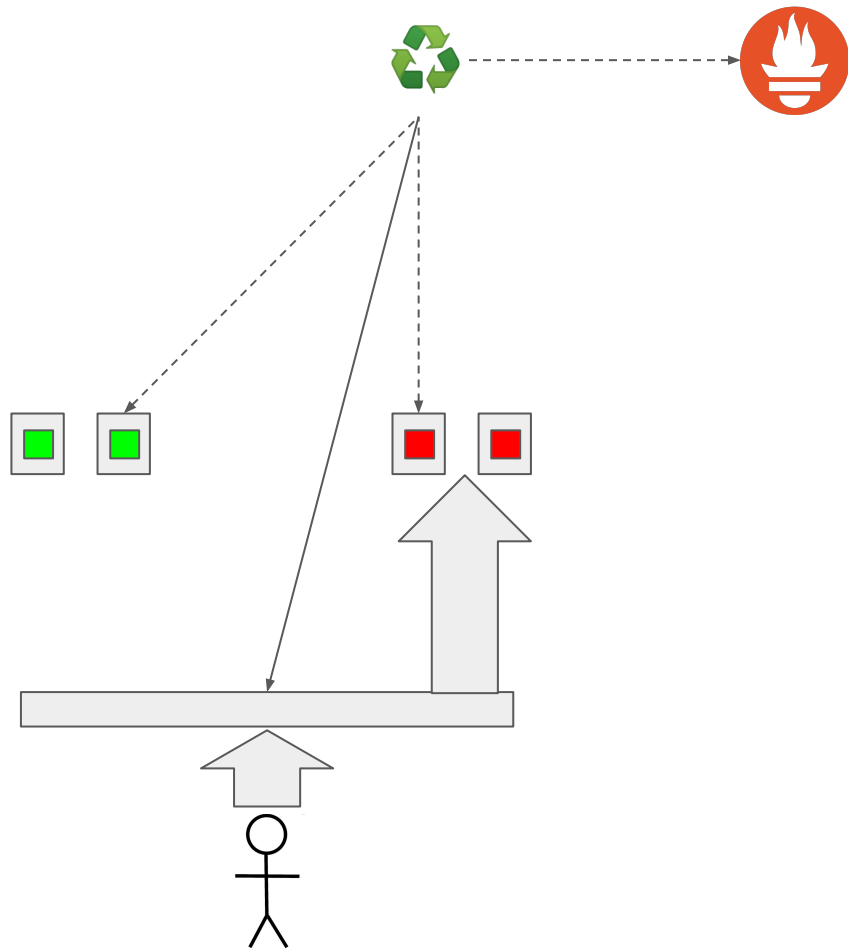
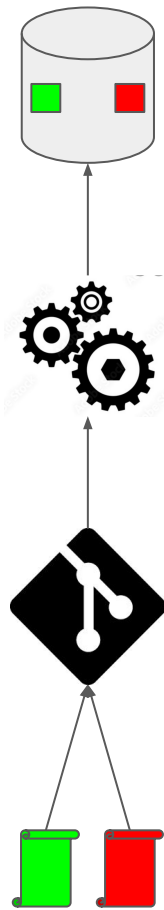
Read-Only

- Gets a mirror of user traffic, but responses dropped
- What's its Service Level? - crash rate, error rate, performance
- Compare results, if helpful

Release







Progressive Roll-Out

- Sends 1% of user traffic to new version
- Monitor all SLIs for a period of time
- If it's within the SLOs, add 1% more traffic

Roll-back

- If it fails SLO at any point, all traffic sent back to the old version
- New version left running for inspection
- Alert raised

Thanks!

@mt165

Slides		mt165.co.uk
Videos		
Demo code		

