CONTINUOUS

ALL THE THINGS

11.08.2016

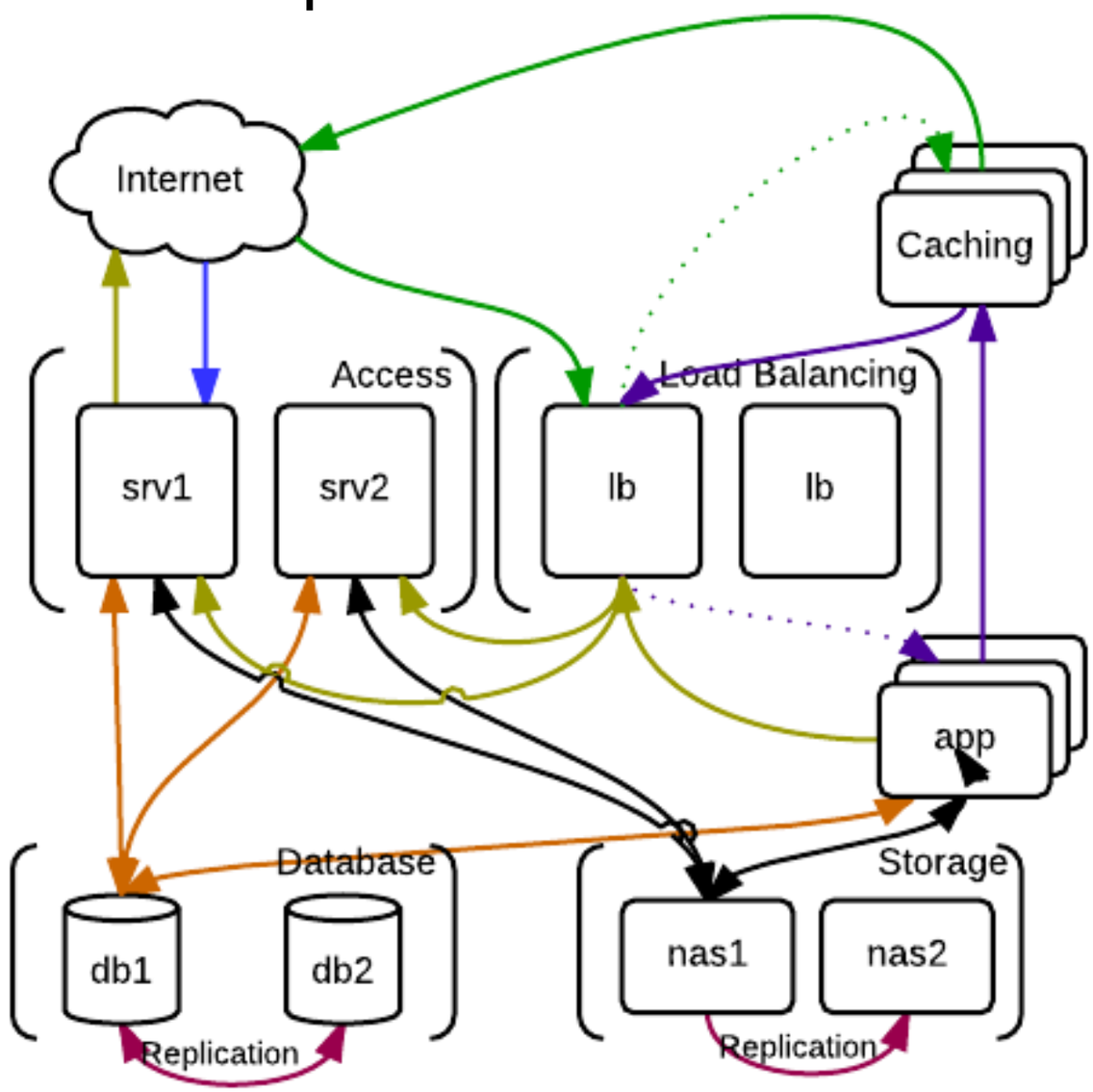
Willie Goosen

CONTINUOUS CHALLENGES

Environment inconsistency: development and test environments that are different to production.

Development Environment

VS.



Production Environment

CONTINUOUS CHALLENGES

Lack of Ownership: development and test environments that are different to production or teams not working close enough.

“It works on my machine…”

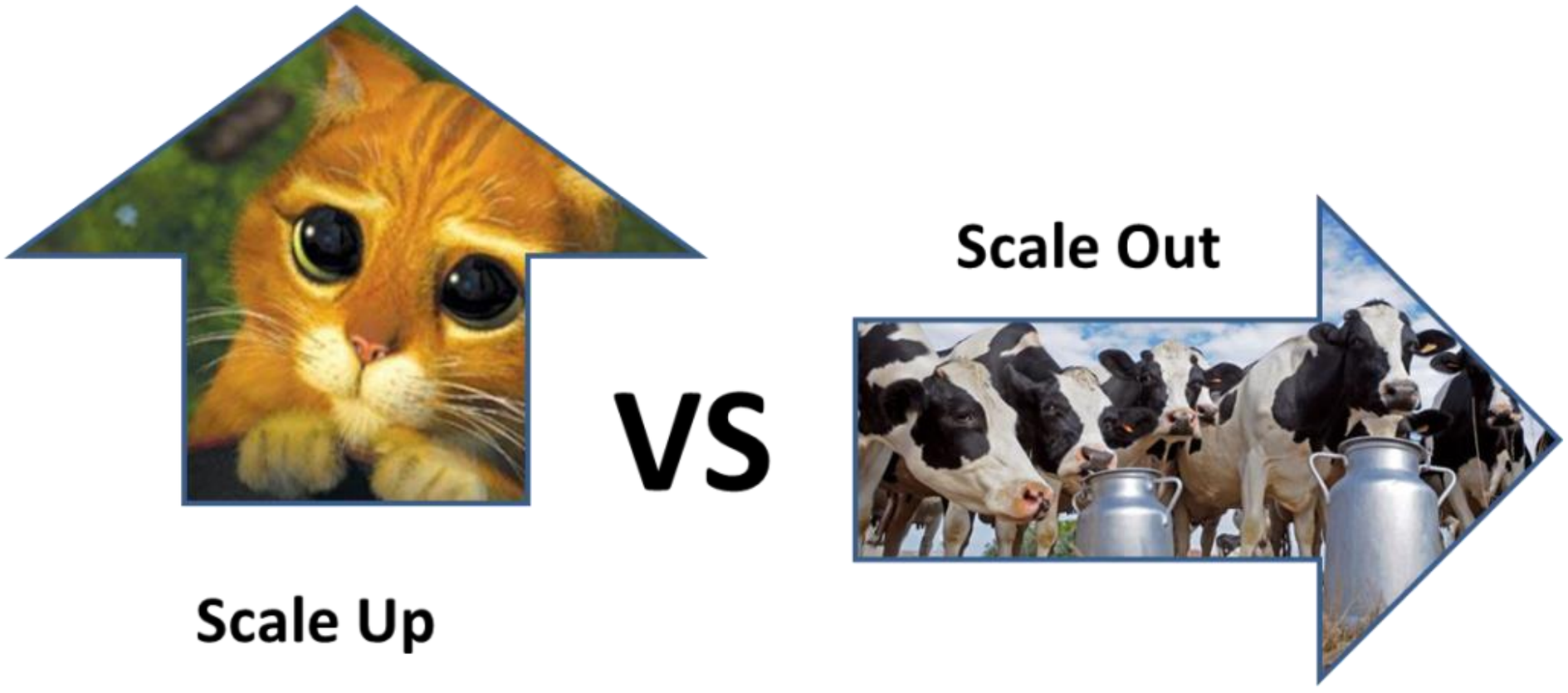
- every developer ever -

“The servers are fine… Must be the code.”

- every sysadmin ever -

CONTINUOUS CHALLENGES

Pets VS Cattle : Treating your servers / services like pets, rather than Cattle, if it gets sick get rid of it.



Name, Love, Nurse VS Numbers, Identical, Disposable

CONTINUOUS CHALLENGES

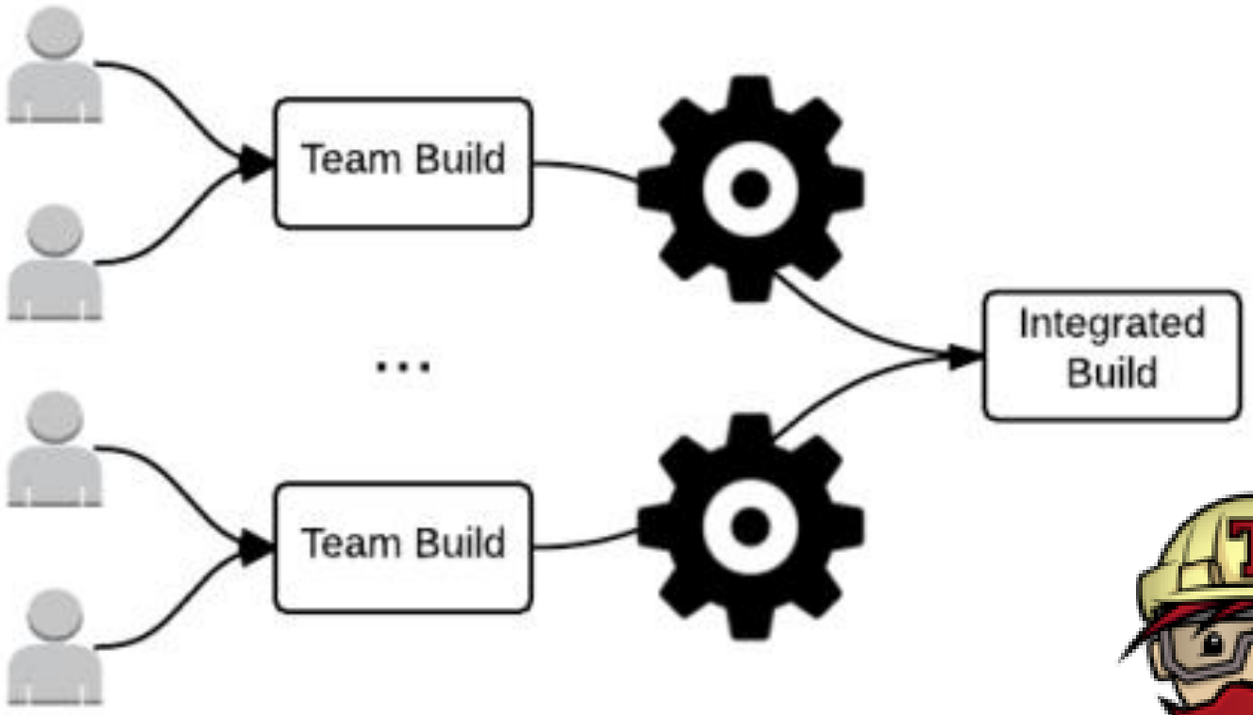
Different Goals : When you have silos of teams with different goal and drivers

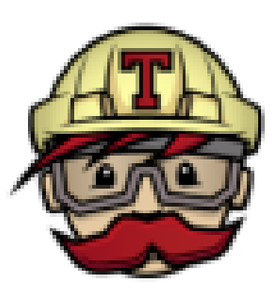
DEV	VS	OPS
• Innovate Through Change		• Generate stability
• Quick turnaround on features		• Rewarded for uptime
• Not concerned of code impact on infrastructure		• Hawk eye on resource consumption of servers

PROBLEM!!

CONTINUOUS INTEGRATION

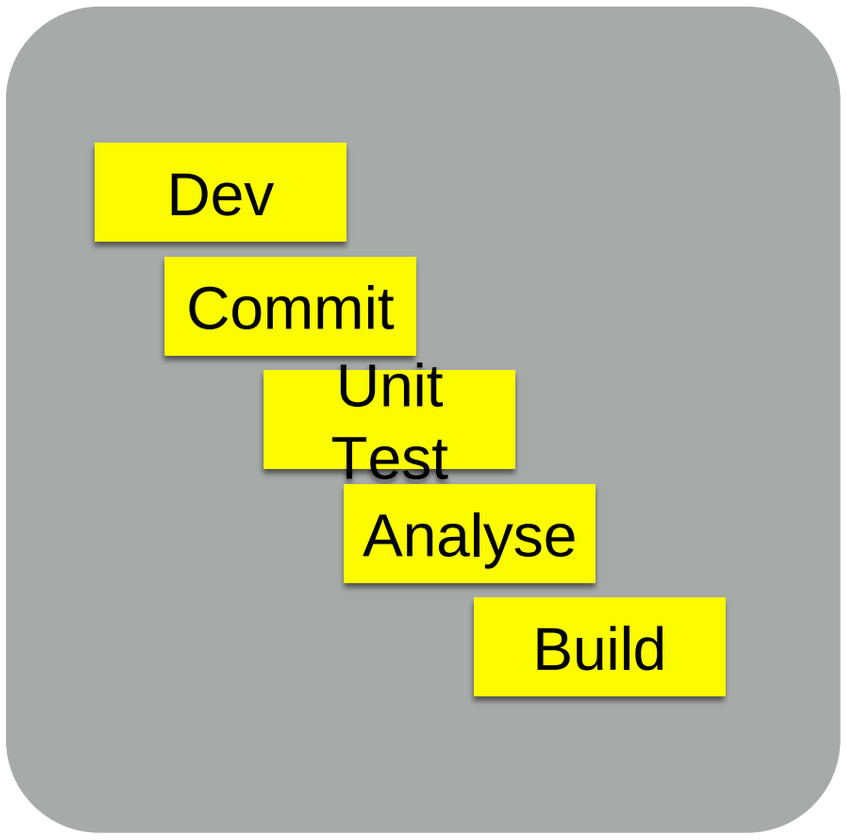
Integrate changes from each developer into the main branch, on checkin / pull request.





CONTINUOUS INTEGRATION

Static Analysis and Unit testing for code quality is performed before any code review happens.



FindBugs™

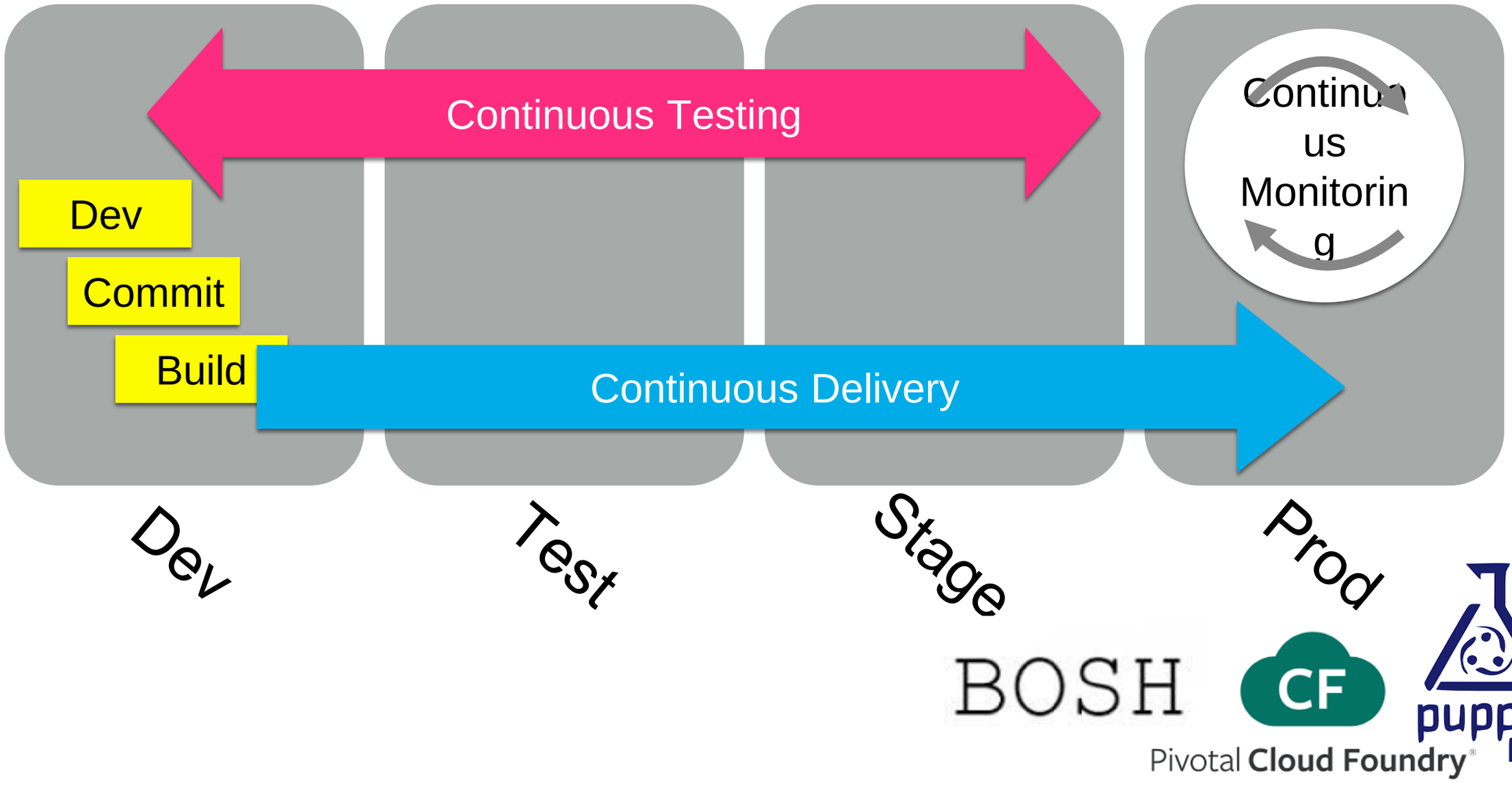
sonarqube

checkstyle

PMD

CONTINUOUS DELIVERY

Deploying the resulting artefact from each Integrated build into various environments



BUZZWORDS

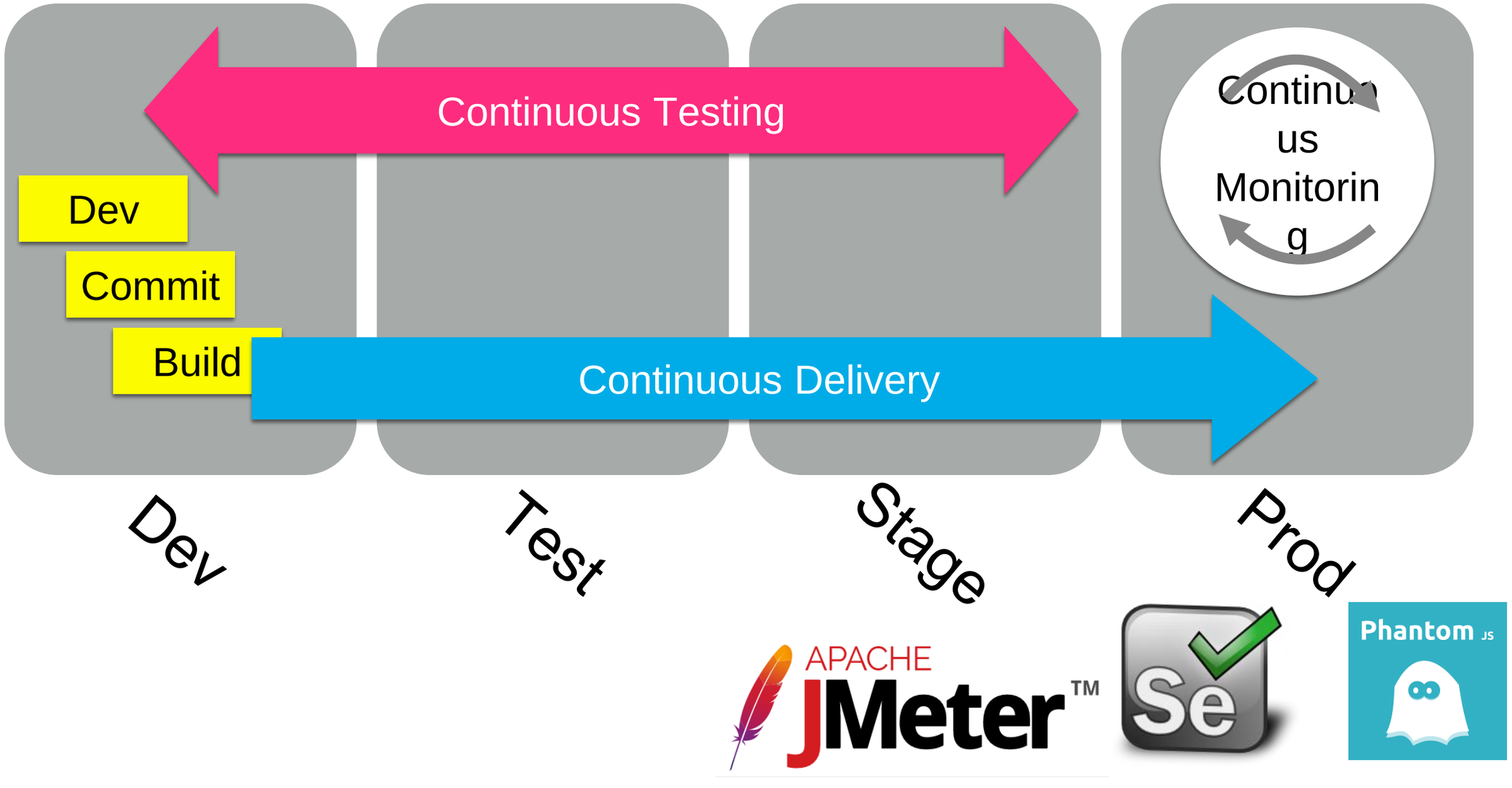
- Immutable Infrastructure - Your infrastructure is treated just like code and checked into source control and build from created from there

- Rolling Deployments - Each server is taken out of the pool and upgraded, then placed back in to service

- Blue Green Deployments - Whole new stack is built, once ready it completely replaces the previous

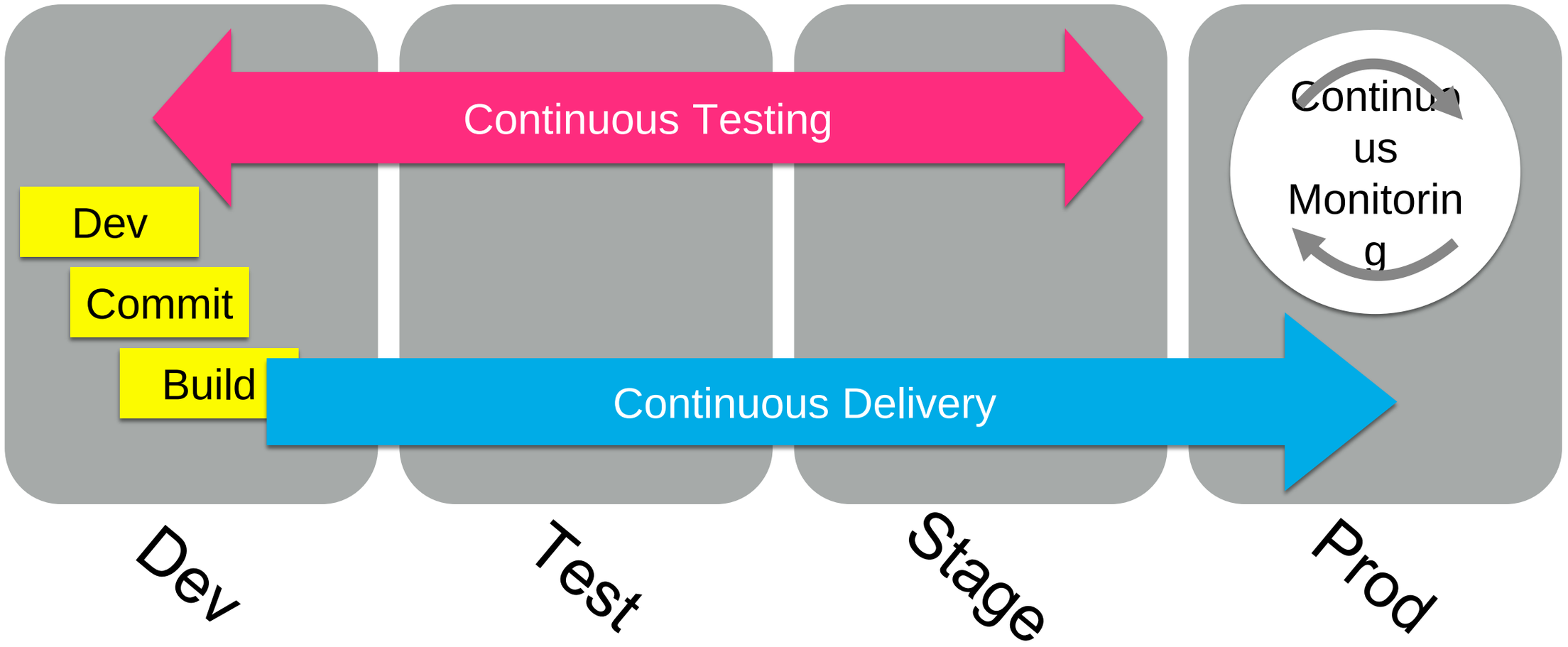
CONTINUOUS TESTING

Depending on the level of test automation, integrated builds can move across environments with or without interaction



CONTINUOUS MONITORING

Ensuring all aspects of each environment is monitored and possibly self-healing



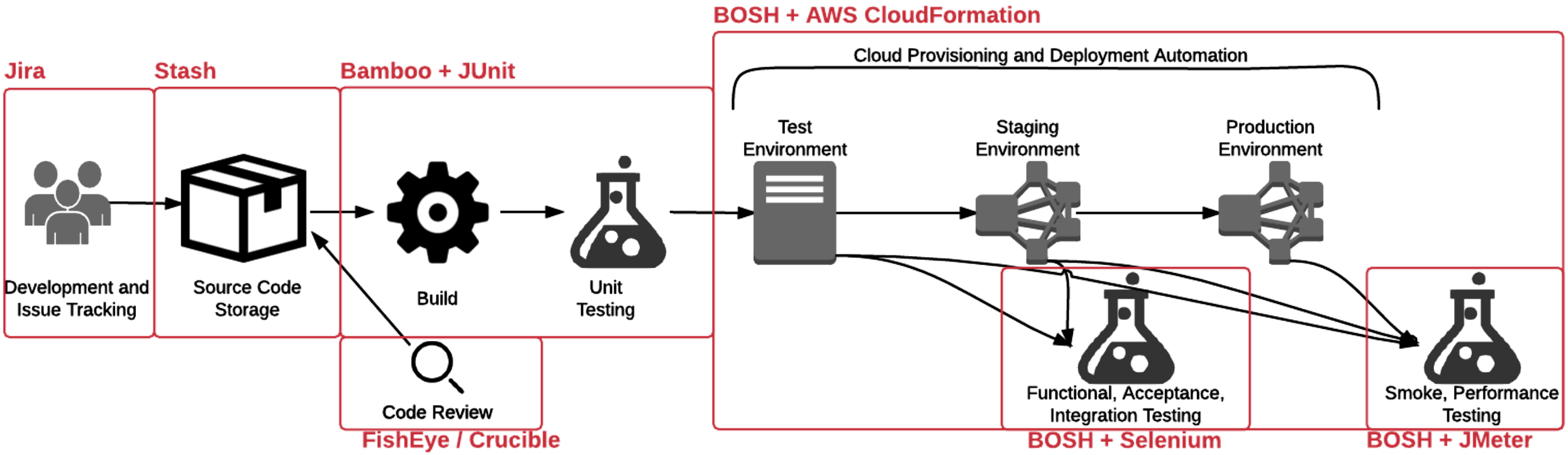
dynatrace ruxit New Relic®

BUZZWORDS

- Application Performance Management - These are specialised tools that inspect the code and interactions with users, web services databases and infrastructure to pinpoint issues.

- Log Aggregation - Central place where aaaaaaallll the logs of a system is stored, usually with a google-for-logs interface to search and find events.

TYING IT TOGETHER



- Ticket Tracking
- Wiki
- Source Code

- Build Tool
- Deployment Tool
- Containers

PARTING THOUGHTS

- Culture - Tighten the interactions between Dev, Test and Ops
- Automation - Establish a practice of automation
- Measure - You can only tune, if you know what the system currently does
- Share - Share the same tools, metrics and data across Dev, Test, and Ops

ASK?!?

ALL THE THINGS