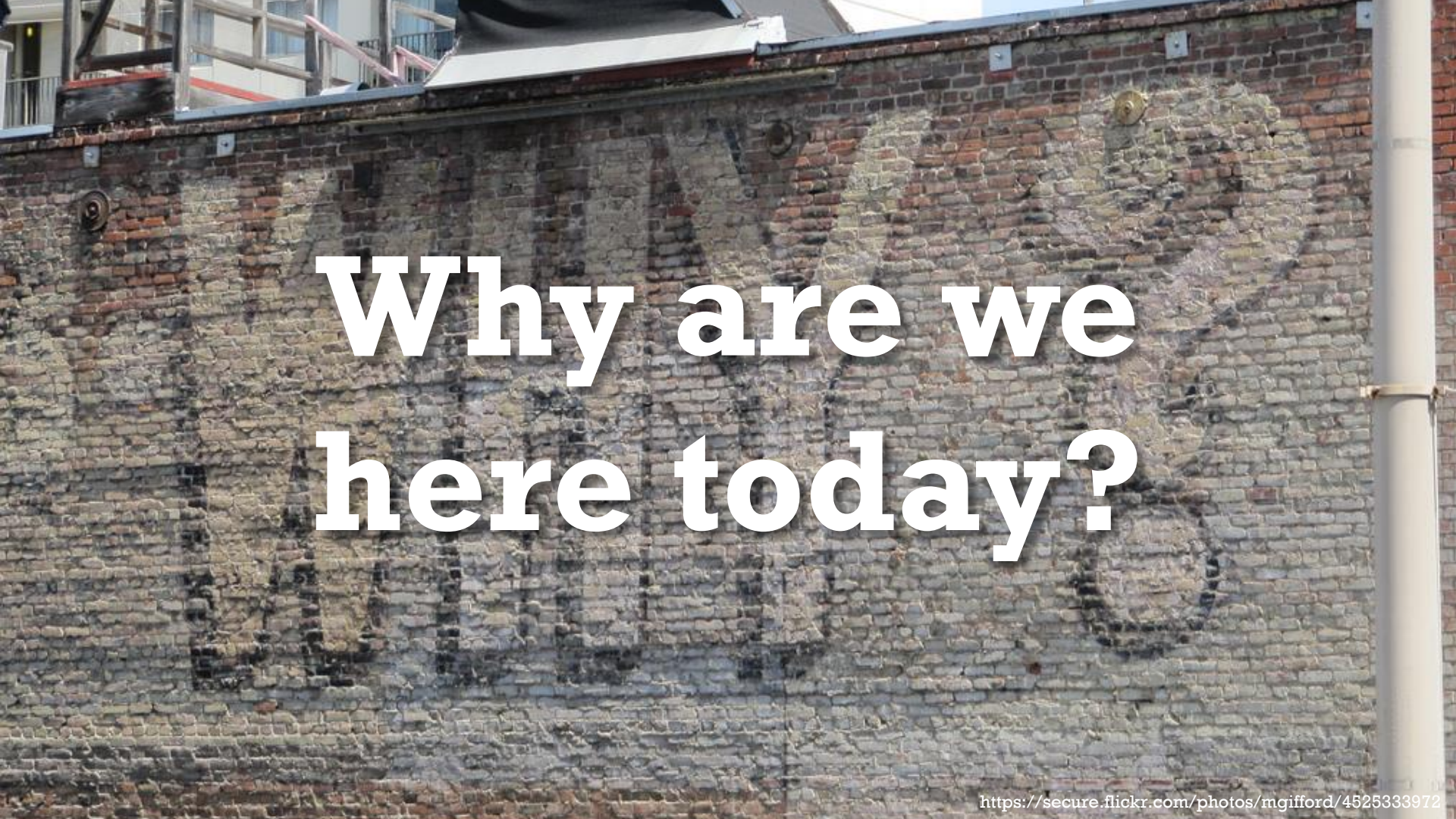
# DevOps @ Amazon:

## Microservices, 2 Pizza Teams, & 50 Million Deploys a Year

Chris Munns – Business Development Manager - DevOps

TriNimbus
Cloud Management Solutions

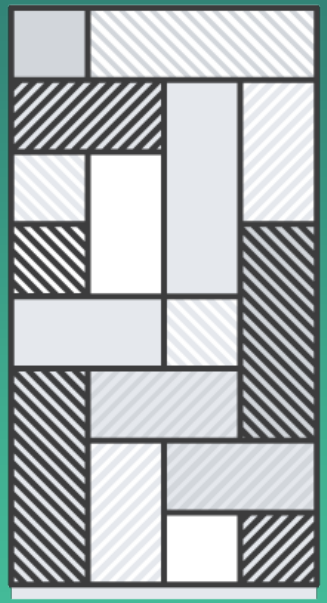
Canadian Executive
Cloud & DevOps Summit




https://secure.flickr.com/photos/mgifford/4525333972


https://secure.flickr.com/photos/pixelthing/15806918992/

# 2001



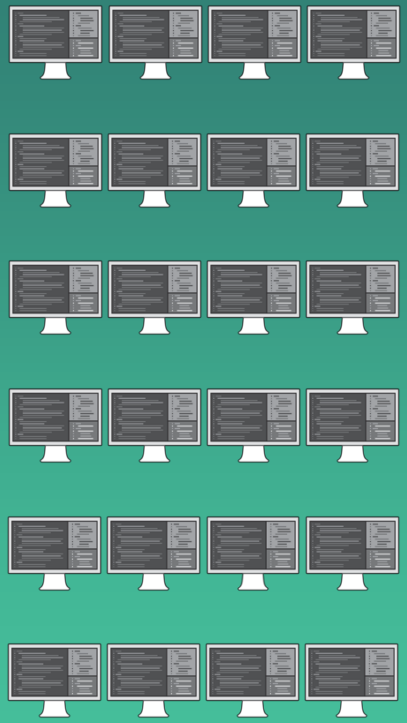
monolithic application

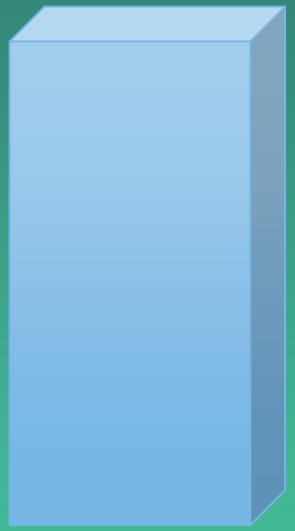
+

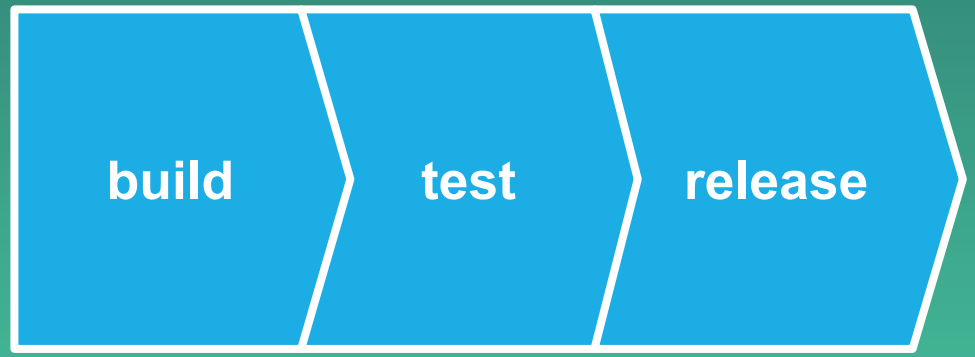
monolithic teams

amazon web services

# Monolith development lifecycle





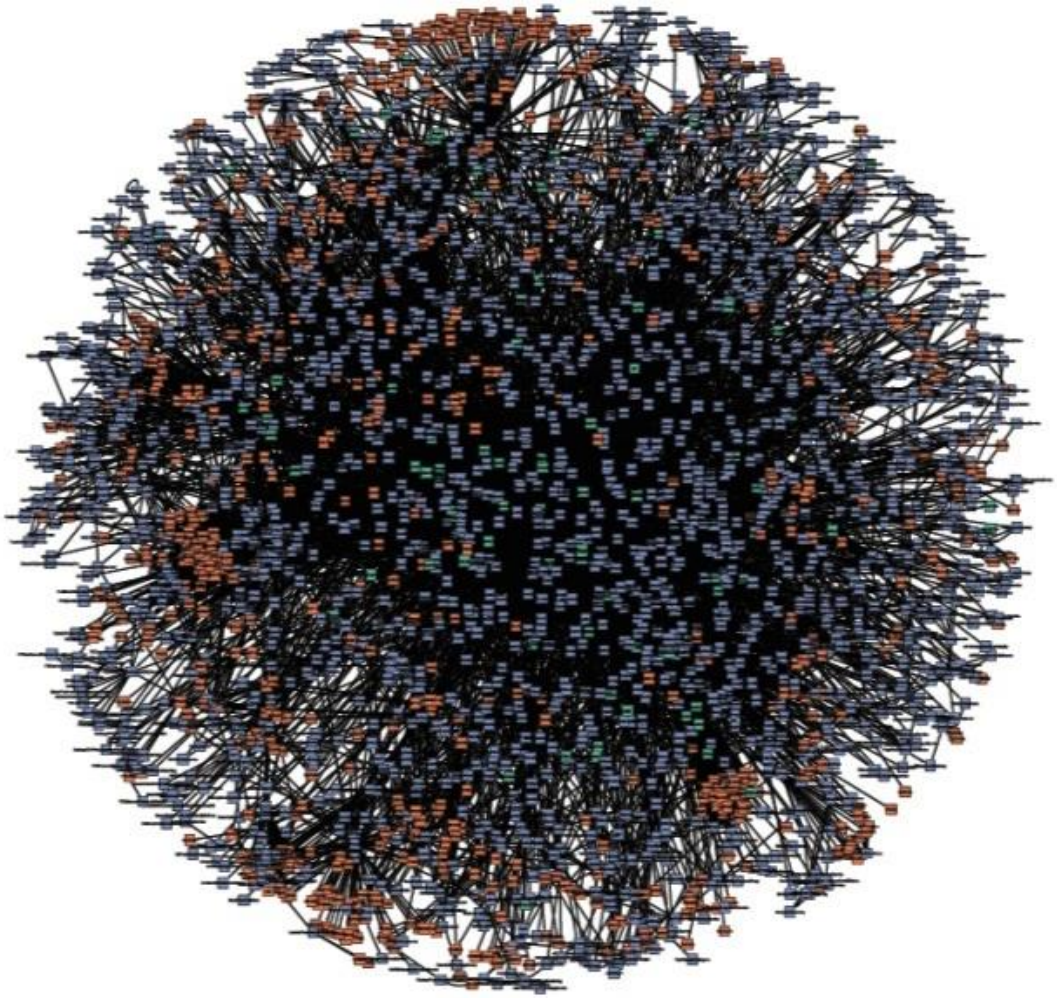



developers

app

delivery pipeline

amazon web services



- Single-purpose
- Connect only through APIs
- Connect over HTTPS
- Largely “black boxes” to each other
- “Microservices”

amazon web services

# Monolithic vs. SOA vs. Microservices

**Monolithic**
Single Unit

**SOA**
Coarse-grained

**Microservices**
Fine-grained

# Microservices vs. SOA

## Microservices:

- Many very small components
- Business logic lives inside of single service domain
- Simple wire protocols(HTTP with XML/JSON)
- API driven with SDKs/Clients

## SOA:

- Fewer more sophisticated components
- Business logic can live across domains
- Enterprise Service Bus like layers between services
- Middleware

amazon web services

- Two-pizza teams
- Full ownership
- Full accountability
- Aligned incentives
- “DevOps”

amazon web services

# How do Two Pizza Teams work?

We call them "Service teams"

- Own the "primitives" they build:
  - Product planning (roadmap)
  - Development work
  - Operational/Client support work
- "You build it, you run it"
- Part of a larger concentrated org (Amazon.com, AWS, Prime, etc)

amazon web services

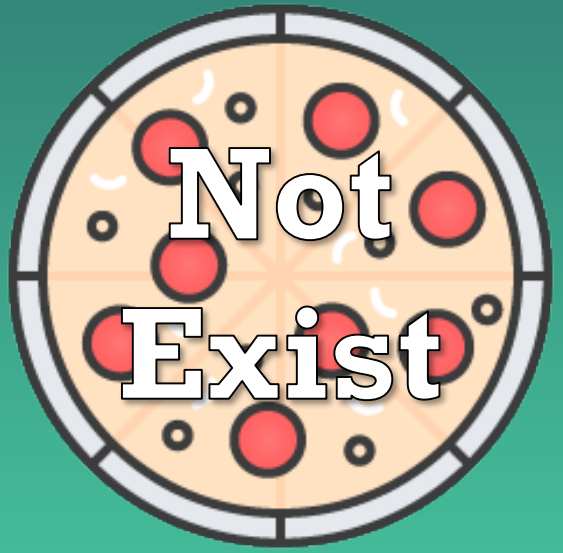
# Who Does QA?


amazon web services


# Who Does On Call?

Image By: Chris Munns - cmunns@amazon.com

# What does Ops Do?



# What about Ops/QA/Etc?

Everyone exists on a "service team" focused on their primitive(s):

- SDE's focused on developing
- PM's focused on product direction
- TPM's help drive development
- SE's focused on infra/tooling
- SDET's focused on test excellence throughout the organization

} Most "2 pizza" teams are just these 2 roles

Some folks are shared across the org, some on individual teams

amazon web services

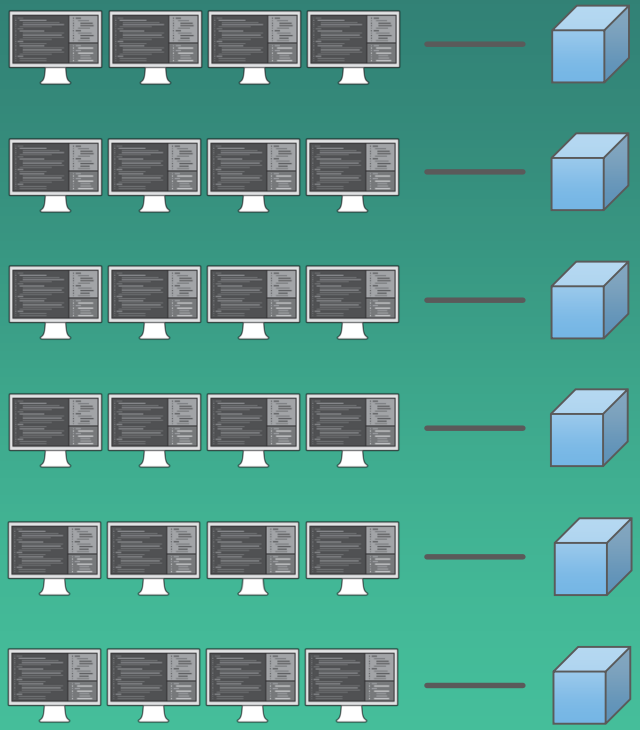
# Boy, that sounds like a lot of freedom?

It is! Teams are empowered and also held to high standards:

- Thorough onboarding/training
- Patterns/practices defined at scale and with 20+ years of organizational knowledge
- Regular technical and business metric reviews
- Regular sharing of new tools, services, technologies, etc, by internal subject matter experts
- Public sharing of COEs; "Correction of Errors" our post-mortem process/tool

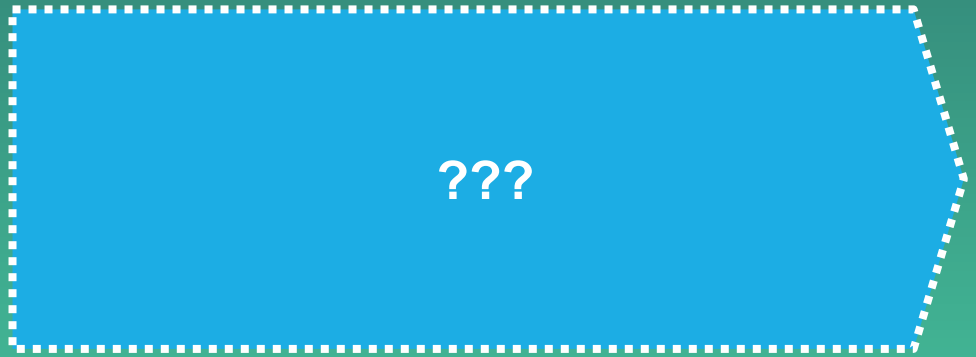
amazon web services

# Missing tools



developers

services




delivery pipeline

amazon web services

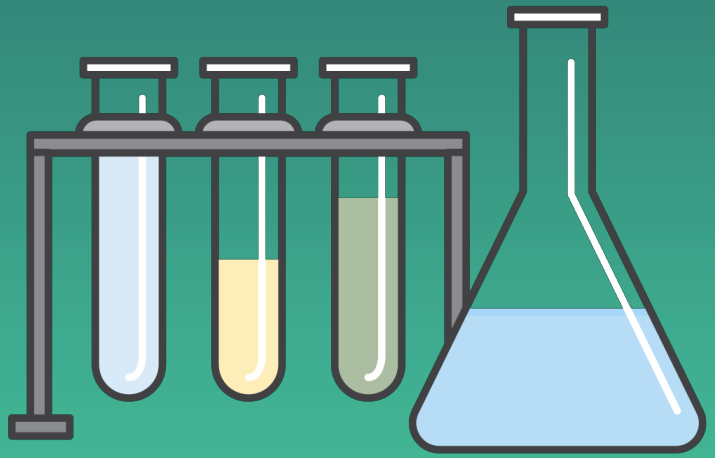
- Self-service
- Technology-agnostic
- Encourage best practices
- Single-purpose services

amazon web services


- Deployment service
- No downtime deployments
- Health checking
- Versioned artifacts and rollbacks

amazon web services

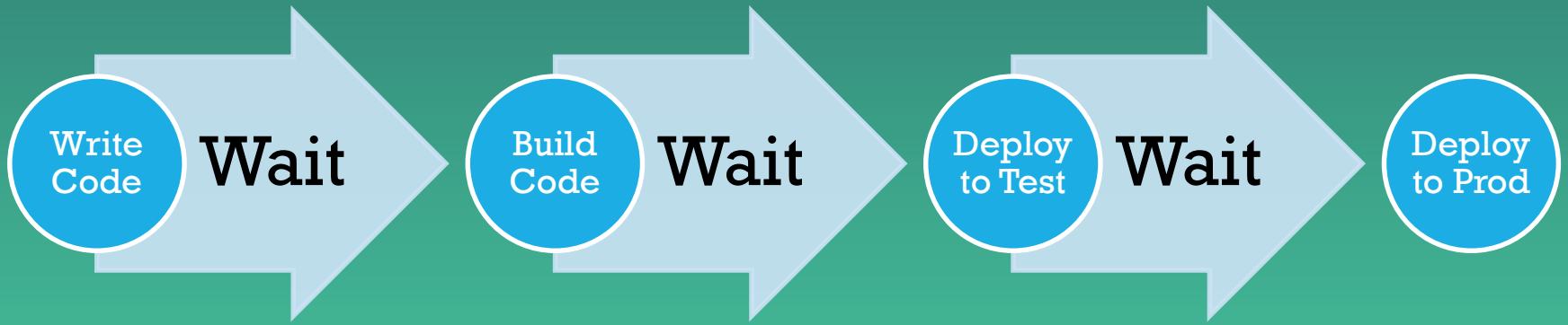
Things went much better under this model and teams were developing features faster than ever, but we felt that we could still improve.



In 2009, we ran a study to find out where inefficiencies might still exist

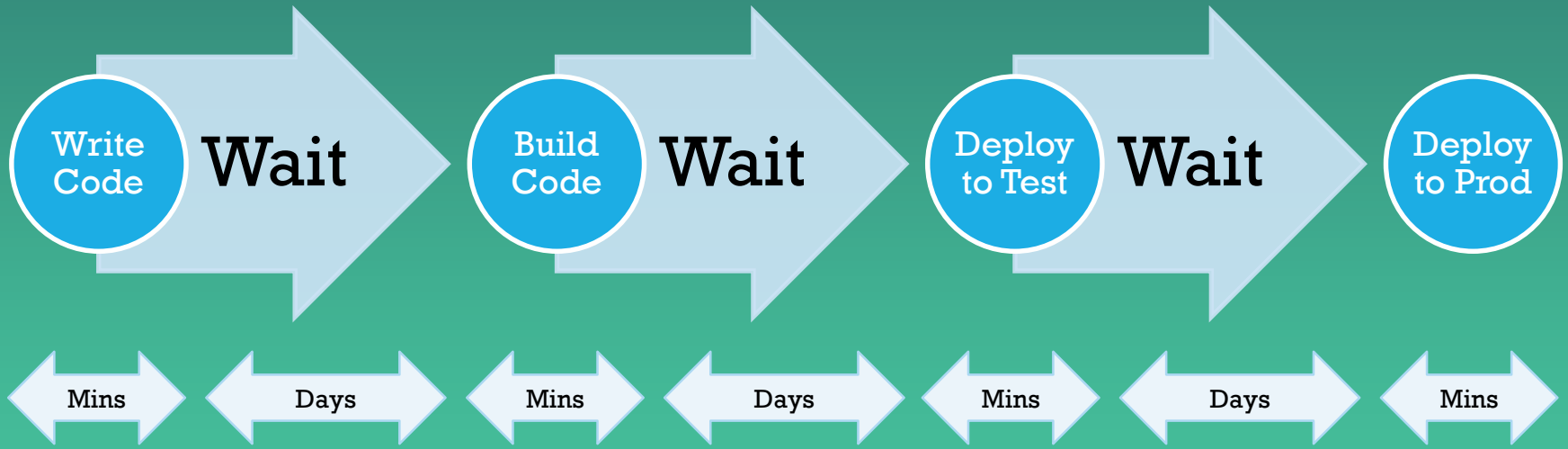
amazon web services

# We were just waiting.



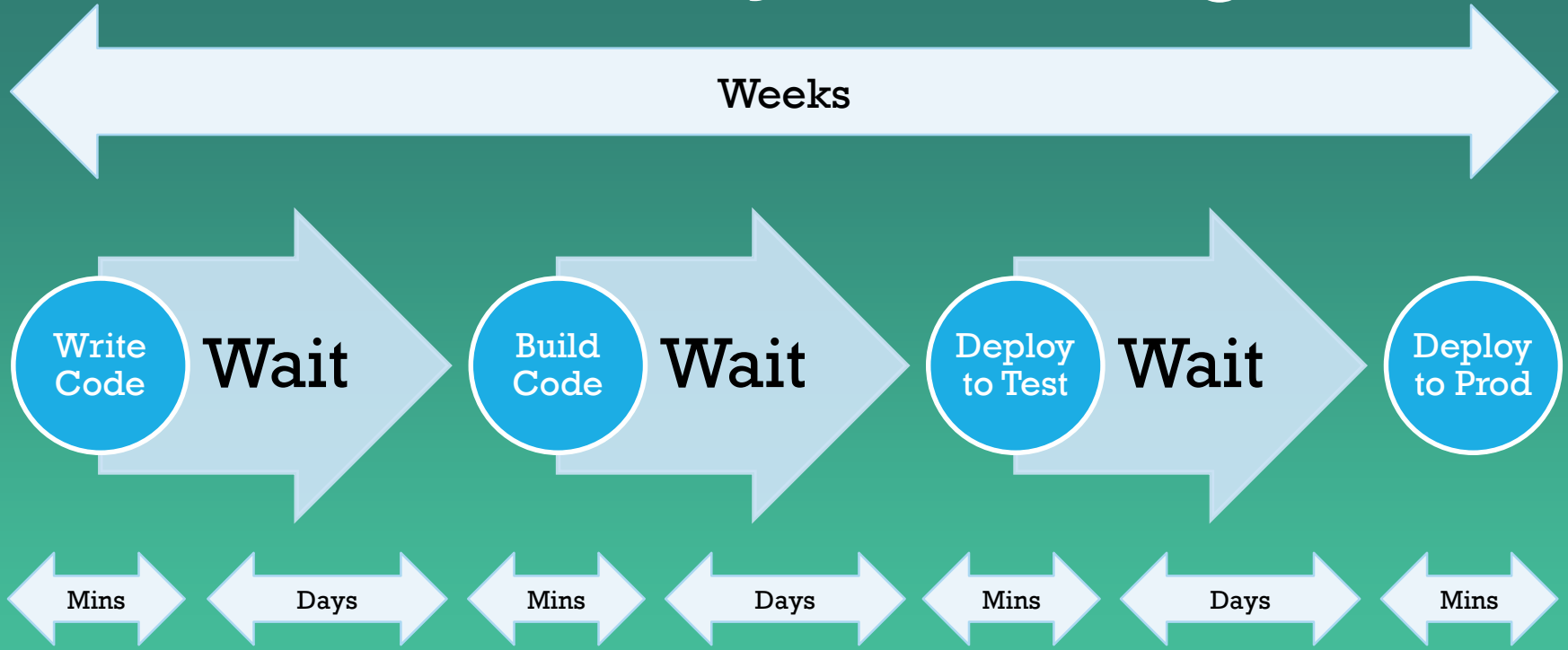

amazon web services

# We were just waiting.

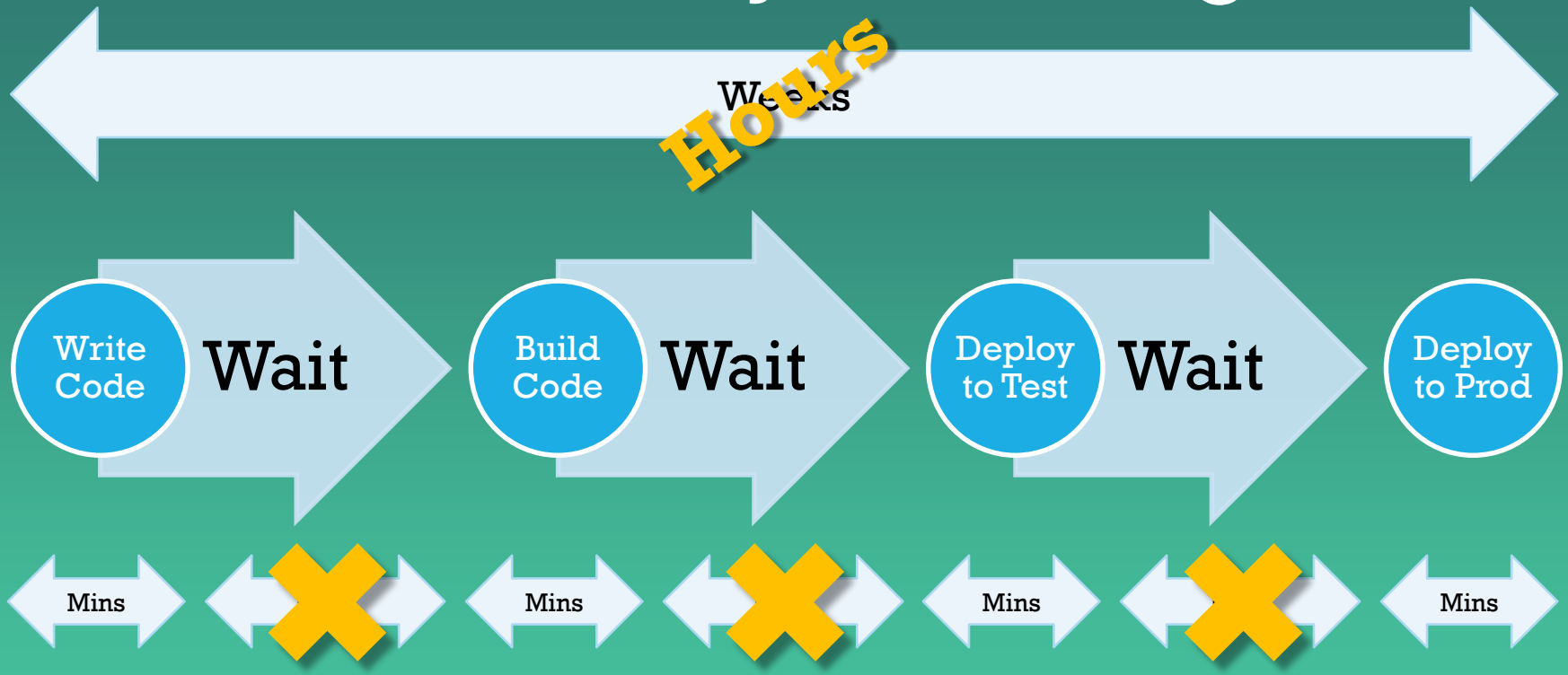



amazon web services

# We were just waiting.



# We were just waiting.




amazon web services

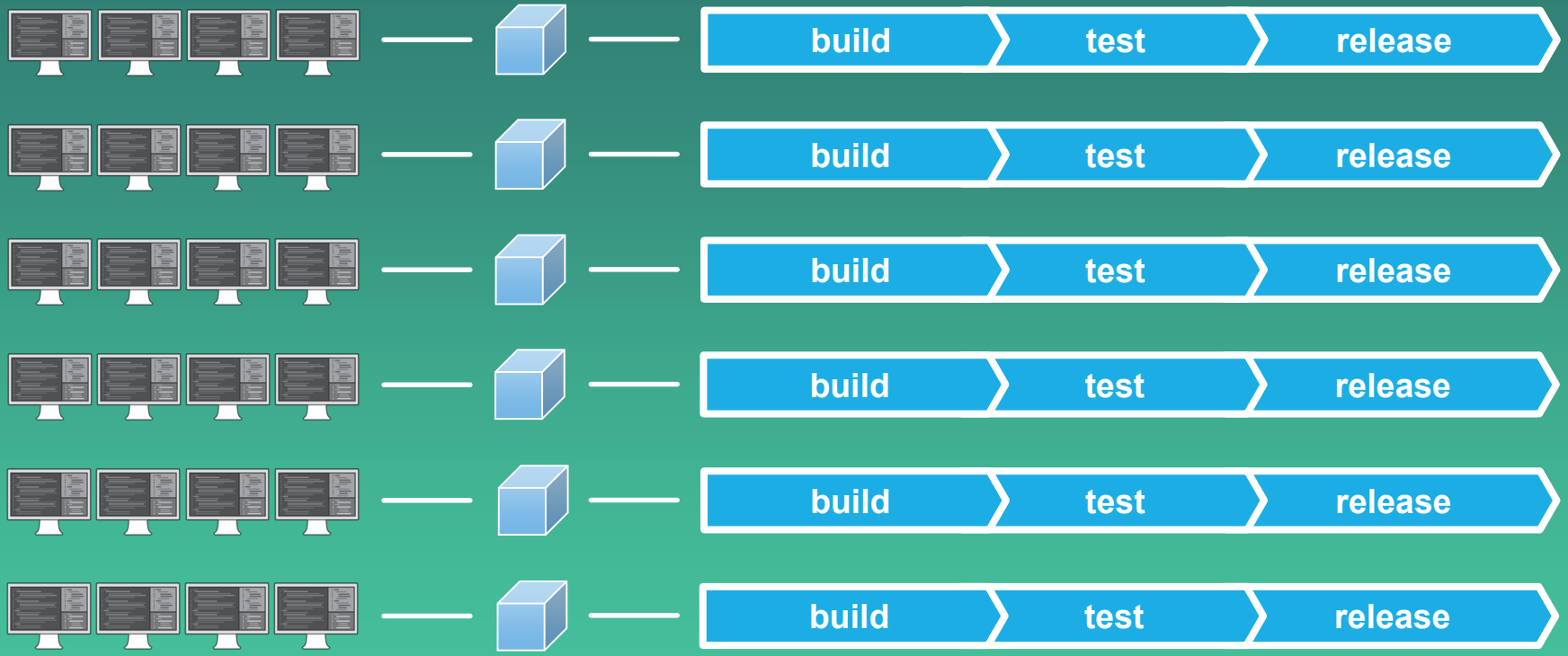

Automated actions and transitions; from check-in to production

Development benefits:

- Faster
- Safer
- Consistent & Standardized
- Visualization of the process

amazon web services

# Microservice development lifecycle




developers services delivery pipelines

amazon web services

# This has continued to work out really well:

- Every year at Amazon, we perform a survey of all our software developers. The 2014 results found only one development tool/service could be correlated statistically with happier developers:

- Our pipelines service!

**continuous delivery == happier developers!**

amazon web services

Thousands of teams

× Microservice architecture

× Continuous delivery

× Multiple environments

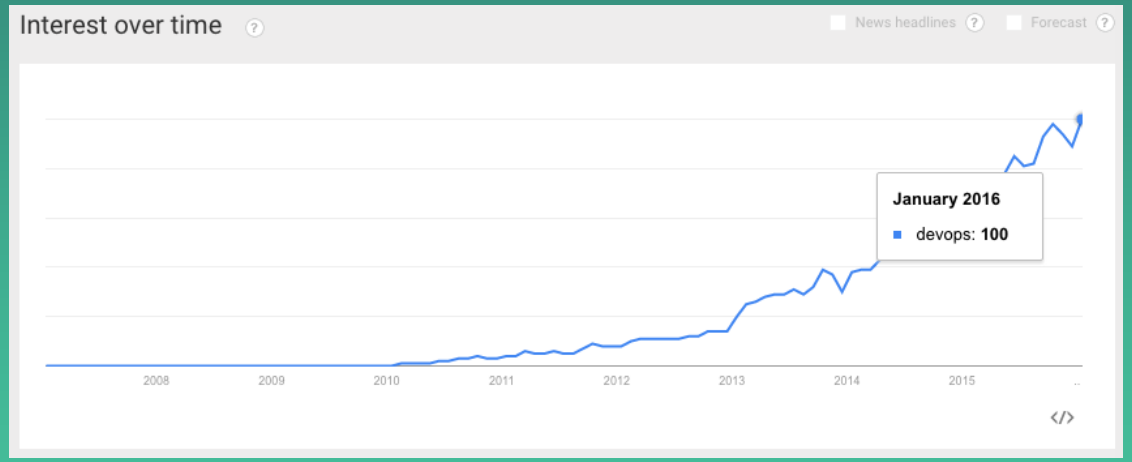
---

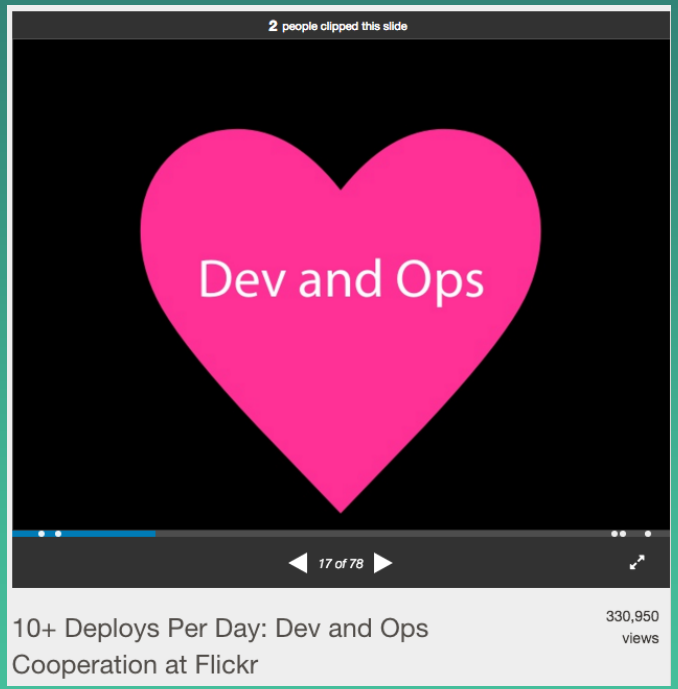
= 50 million deployments a year*

*2014 number

# Quick aside on DevOps

In 2009 (same year as our study) the term "DevOps" first started to appear:




Source: Google Trends & http://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr/

amazon web services

# What is DevOps?

Cultural Philosophy + Practices + Tools

amazon web services

# What is DevOps?

Cultural Philosophy

Practices

Tools

# What is DevOps?

## Cultural Philosophy

## Practices

## Tools

- Tearing down barriers
  - Between teams
  - Mid-process
- Enable the smart people you are spending time and money hiring to make smart decisions
- Assigning ownership, accountability, responsibility to the people doing the work, aka “you build it, you run it”
- Reducing responsibility to the most directly involved individuals
- Increase visibility to the big picture and the results of work being done

amazon web services

# What is DevOps?

Cultural Philosophy

Practices

Tools

- Continuous Integration
  - Application testing/QA work applied throughout the development
- Continuous Delivery
  - Automated deployment capabilities of code across environments
- Infrastructure as Code
  - No hand carved infrastructure
- Self-service environments
  - Remove procurement blockers for basic needs
- Microservices
  - Break down complicated monolithic applications in to smaller ones

# What is DevOps?

Cultural Philosophy

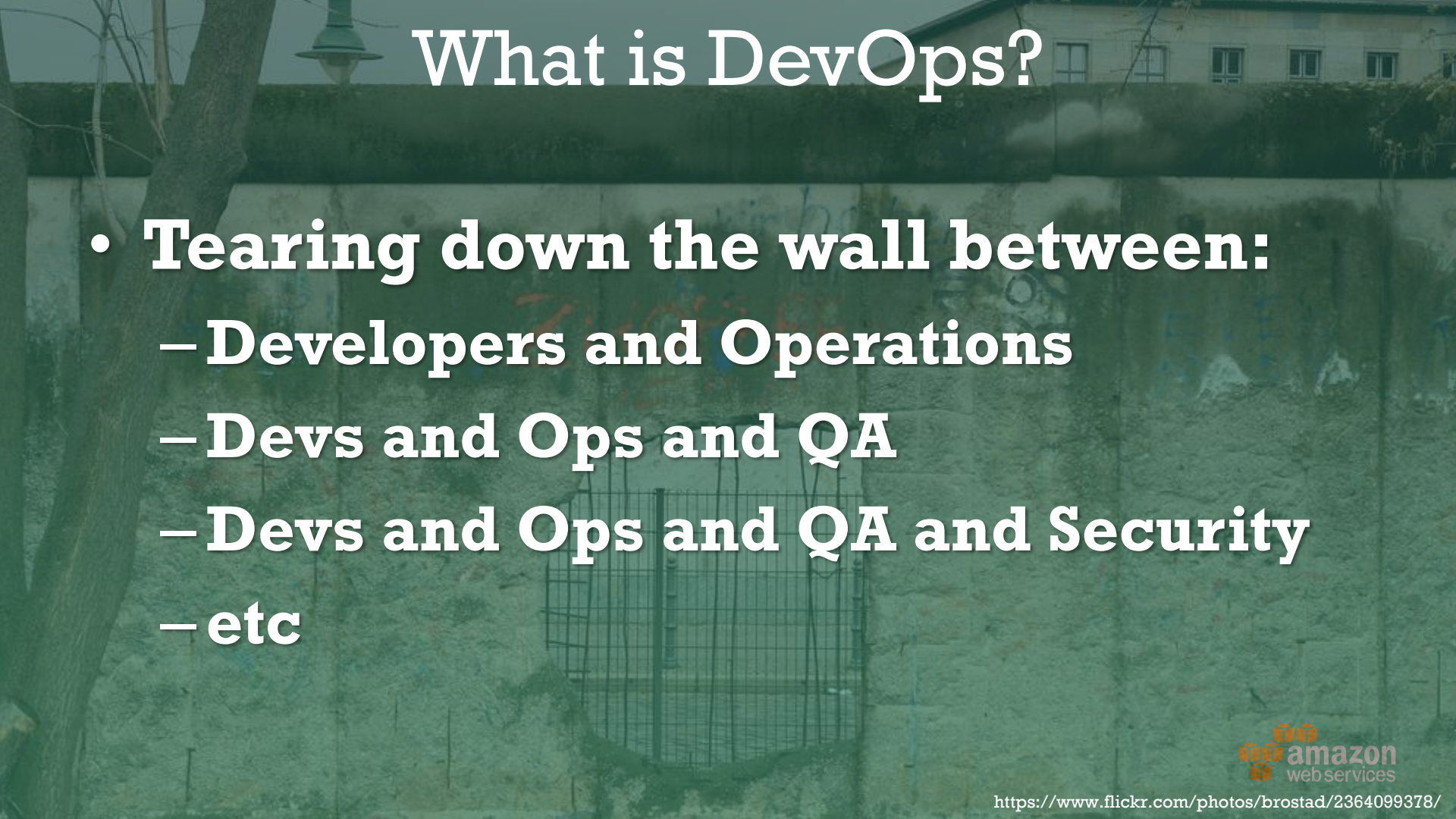
Practices

Tools

- Automated development pipeline tooling
  - Application testing frameworks
  - Code review/feedback tools
  - Automated static analysis
- Consistent and predictable application management & configuration management tools
- Consistent infrastructure measurement tools
  - Metrics
  - Logging
  - Monitoring
  - APM
- Security analysis and management tools

# What is DevOps?

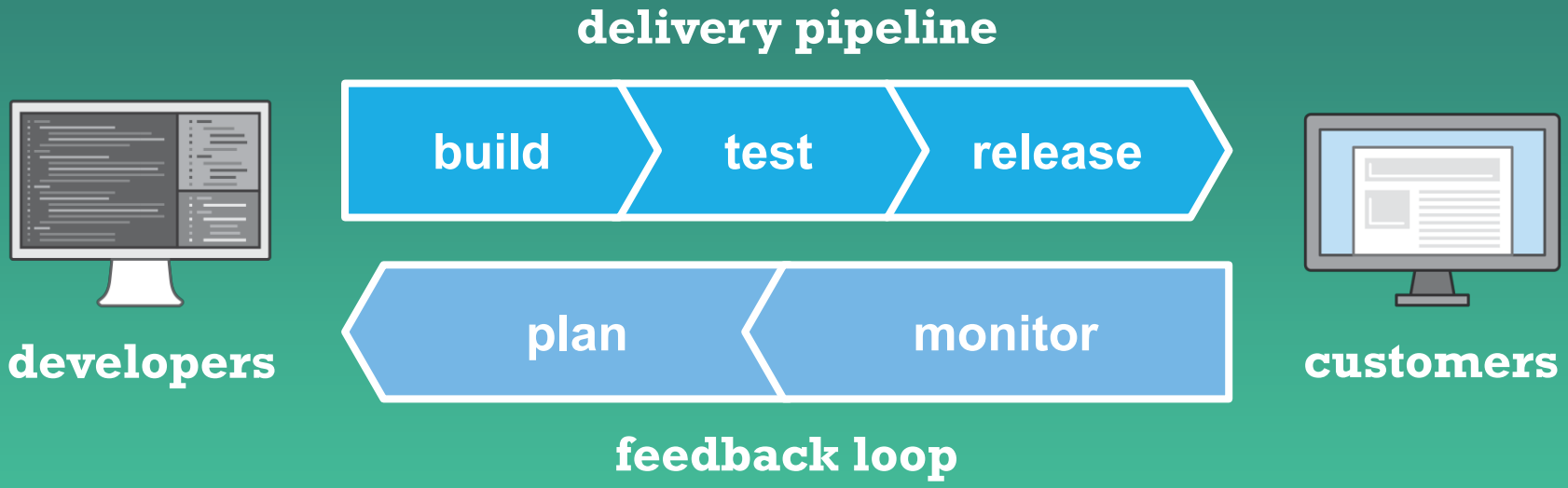
- **Tearing down the wall between:**
  - **Developers and Operations**
  - **Devs and Ops and QA**
  - **Devs and Ops and QA and Security**
  - **etc**



amazon web services

https://www.flickr.com/photos/brostad/2364099378/

# What is DevOps?




DevOps == efficiencies that speed up this lifecycle

amazon web services

# DevOps – Don't Take It Just From Us:

- **Oscar Health**: 2 systems engineers, 45+ Developers, self service infrastructure tools, HIPAA requirements
- **AirBNB**: 5 Operations people for 1000+ instances
- **Gilt**: several hundred microservices, self service tools extending AWS, almost entirely on t2 instances
- **Intuit/TurboTax**: "deployed over 40 simultaneous experiments during the peak filing season"
- **MLB**: scale up massively during Minor League games and events, turn it off later





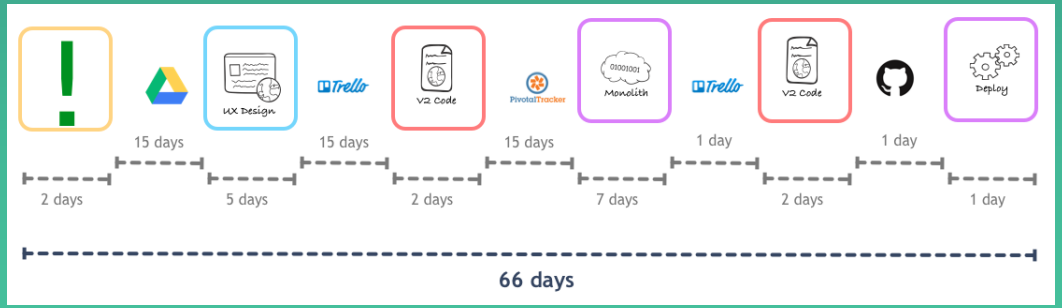
# DevOps – Don't Take It Just From Us:


- Capital One's moved from traditional waterfall to DevOps:
- 100s of code commits per day,
- Integration from once a month to every 15 minutes
- QA from once per month to 4 times per day
- Deployment from manual to completely automated
- Production release from monthly/quarterly to once per sprint

amazon web services

# DevOps – Don't Take It Just From Amazon:

In September 2015 Phil Calcado, ex-SoundCloud, wrote in "How we ended up with microservices." how SoundCloud reduced product development lead times from 66 days to 16!*





*http://philcalcado.com/2015/09/08/how_we_ended_up_with_microservices.html

amazon web services

# 5 Key technology areas to focus on:

# 5 Key technology areas to focus on:

Why these 5?

- Typically under-invested areas for most organizations
- Areas where developers/operations folks are constantly reinventing the wheel
- Important key components of DevOps technology and practices
- Mastering these areas will lead to better:
  - development velocity
  - reduced RTO/RPO
  - better cost management
  - reduction in ”fire from the hip” data-less infrastructure/application tweaks
  - improved security/governance/compliance

# 5 Key technology areas to focus on:

- Continuous Integration/Delivery
- Infrastructure as Code
- Monitoring/Metrics/Logging/APM
- APIs/Microservices Management
- Communication & Collaboration

## DevOps Technology Partners















## DevOps Consulting Partners














# FIN, ACK

Some take aways:

- By stepping back and looking at the big picture of how we were working we were able to drive numerous improvements over the years
- By automating and *as-a-Service-ing as many work flows as possible we remove the majority of waiting, which was a majority of the overall time
- DevOps: Every company will have their own unique culture but the practices are looking fairly consistent
- DevOps: By investing in tools you can standardize on you remove the burden of developers needing to figure out how they will solve these problems on their own
- DevOps: If culture isn't driven from the top down it rarely will work out


https://secure.flickr.com/photos/dullhunk/202872717/