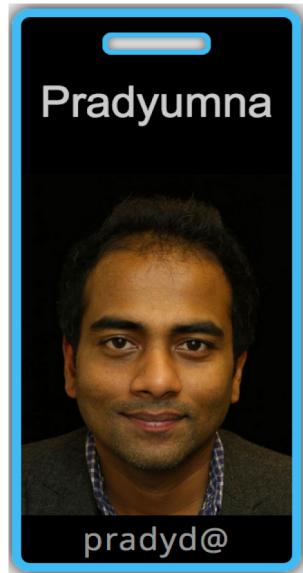


# Introduction to DevOps on AWS



Pradyumna Dash("Prady")

Solutions Architect, Amazon Web Services

15 years infrastructure experience

11 years systems, solutions architecture and DevOps experience

Passionate about Containers, Microservices, AI/ML

[pradyd@amazon.co.uk](mailto:pradyd@amazon.co.uk)

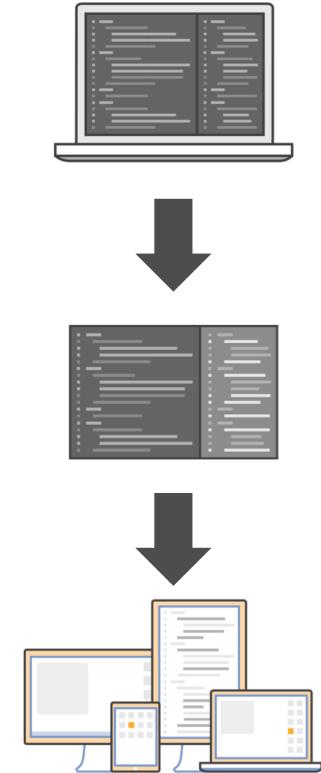
# What we'll cover

- What is DevOps?
- The Amazon DevOps story
- AWS Code Services
- AWS DevOps Portfolio

# Why are we here today?

**Software creation and distribution  
is easier and faster than ever:**

- Startups can now take on giants with little to no funding ahead of time
- Getting your software into the hands of millions is a download away
- Your ability to move fast is paramount to your ability to fight off disruption



# What is DevOps?

# What is DevOps?

- Cultural philosophies
- Practices
- Tools

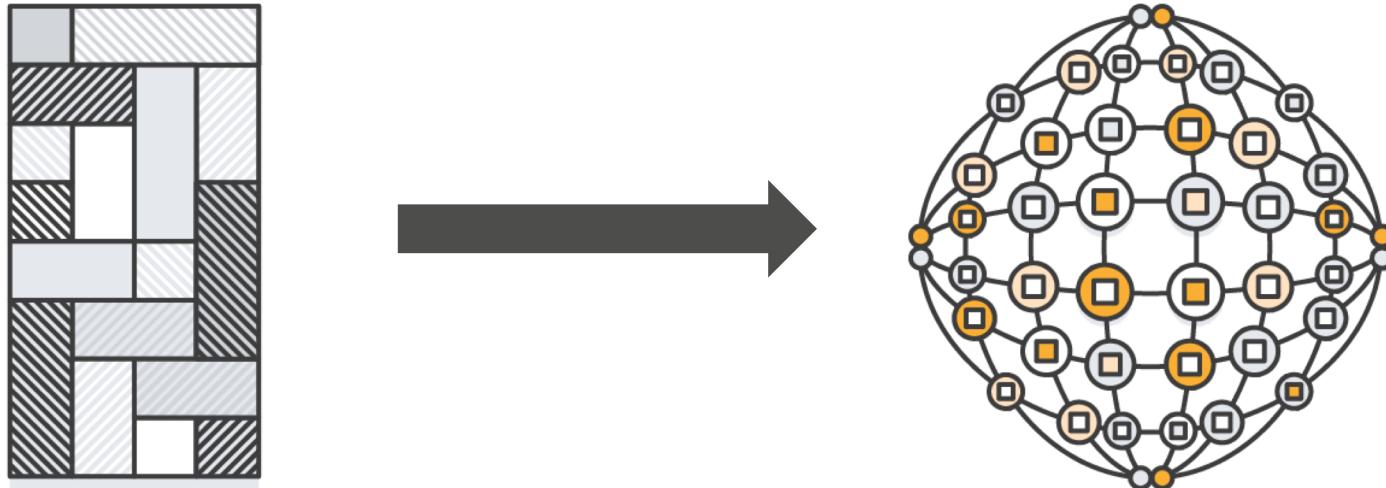
# DevOps Culture

- Dev & Ops coming together
  - No more “silos”
- Shared responsibility
- Ownership
- Visibility and communication

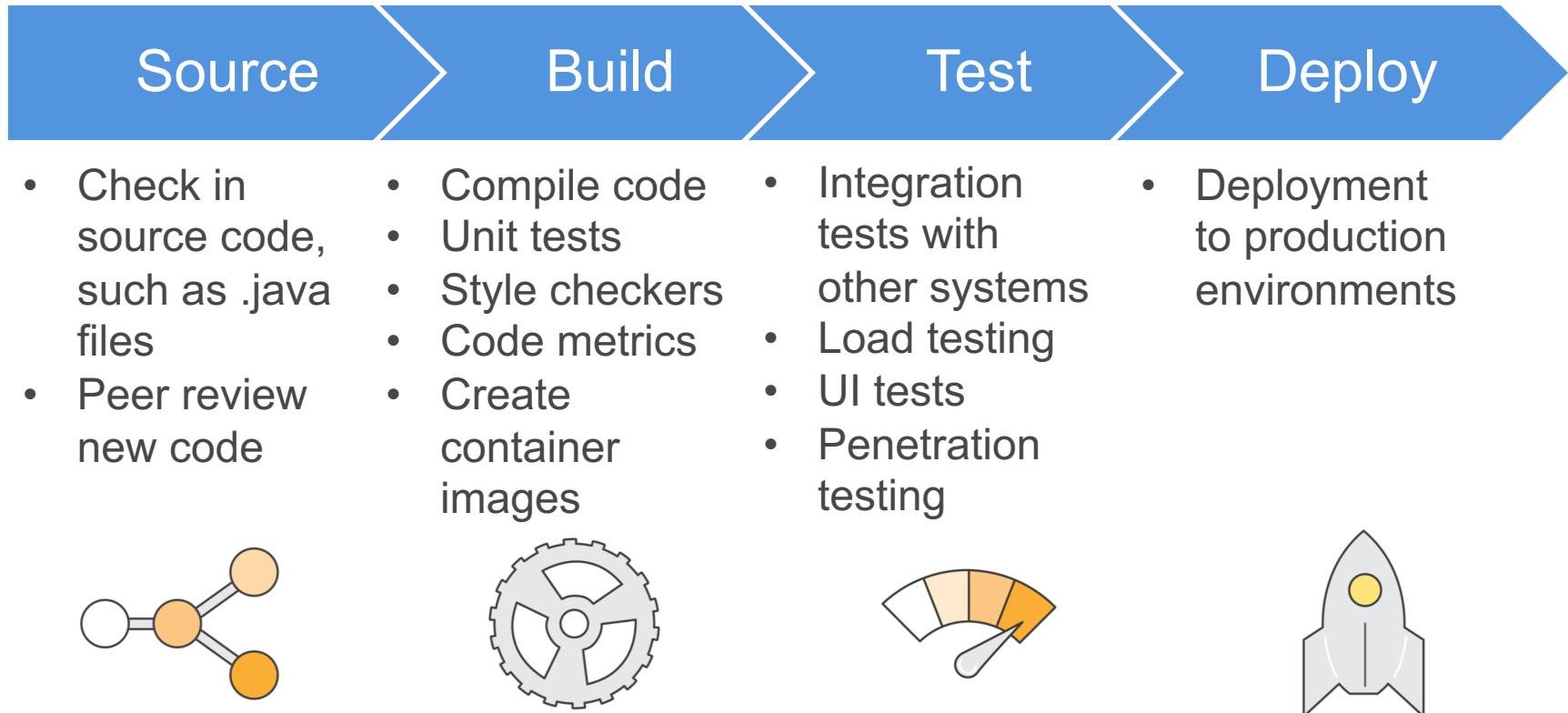


# DevOps Practices

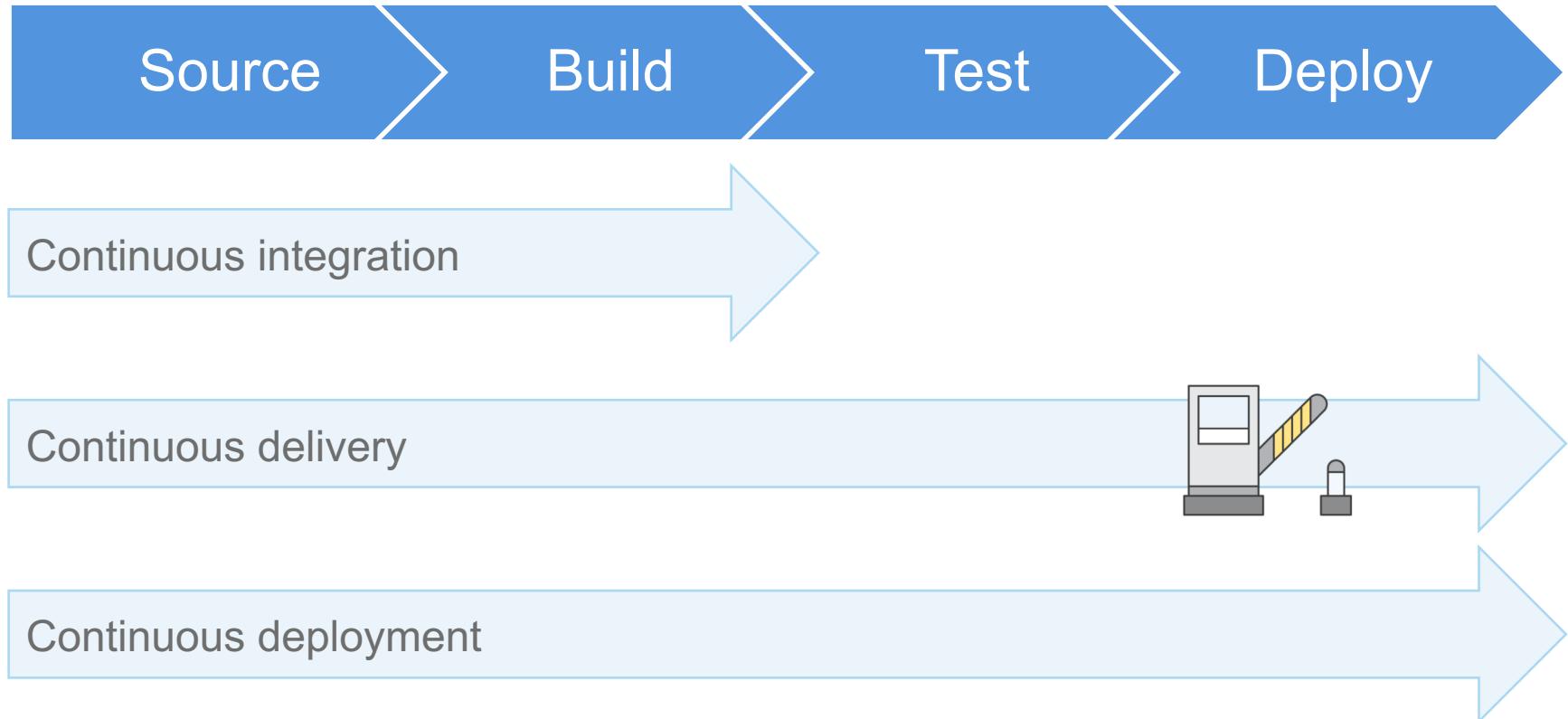
- Microservices architecture
  - Moving away from large “monolithic” architectures to smaller decoupled services



# Release process have for major phases



# DevOps Practices



# Enabling common DevOps practices

- Infrastructure as code
- IT automation
- Continuous integration
- Continuous deployment
- Integrated version control (application and infrastructure)
- Monitoring and logging

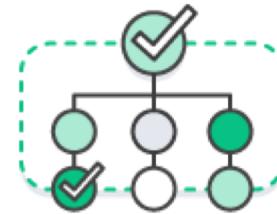
# Benefits of DevOps



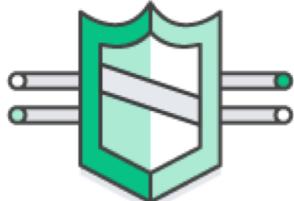
Improved Collaboration



Rapid Delivery



Reliability



Security



Scale



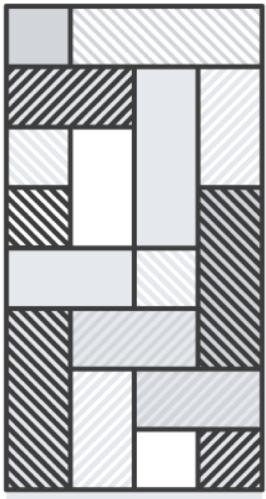
Speed



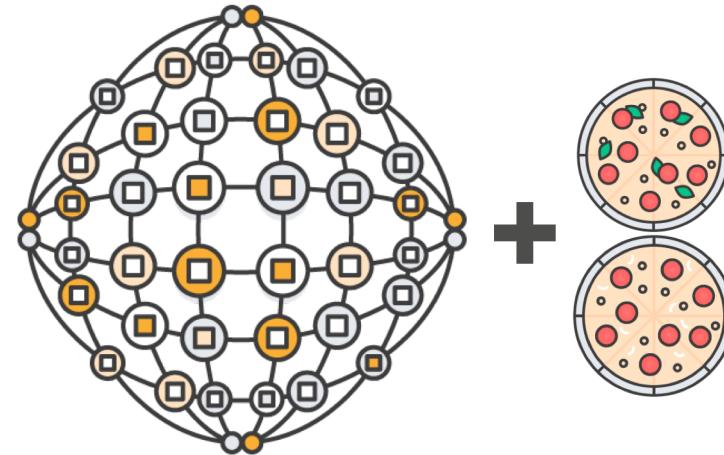
# A look back at development at Amazon...

# Development transformation at Amazon: 2001-2009

2001



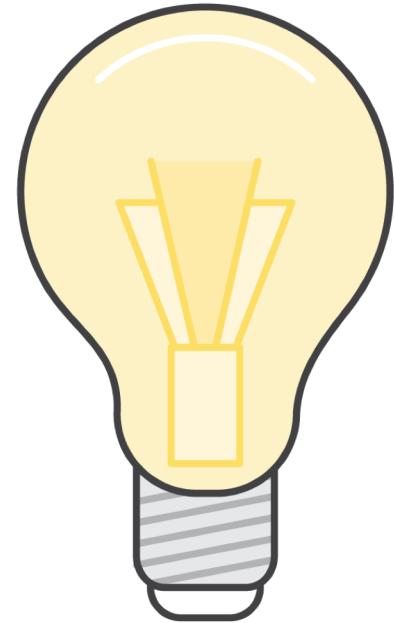
2009



monolithic architecture +  
hierarchical organization

Decoupled services +  
2 pizza teams

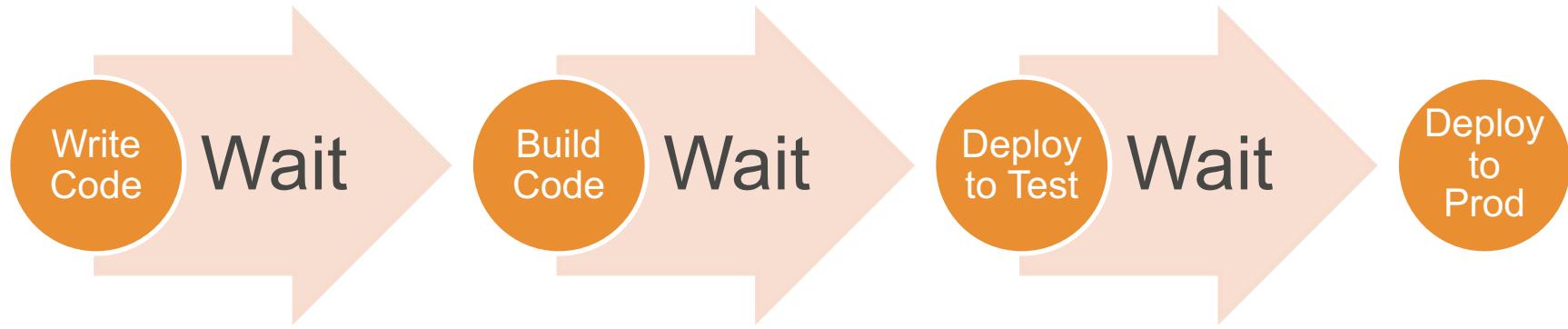
- Things went much better under this model and teams were releasing faster than ever, but we felt that we could still improve.



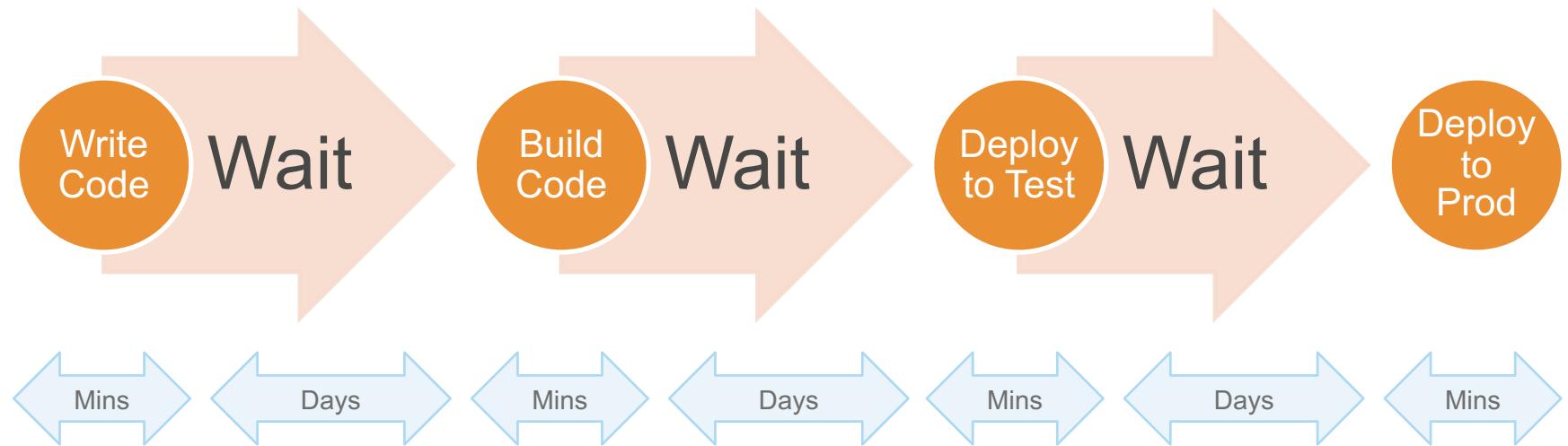


- In 2009, we ran a study to find out where inefficiencies might still exist.

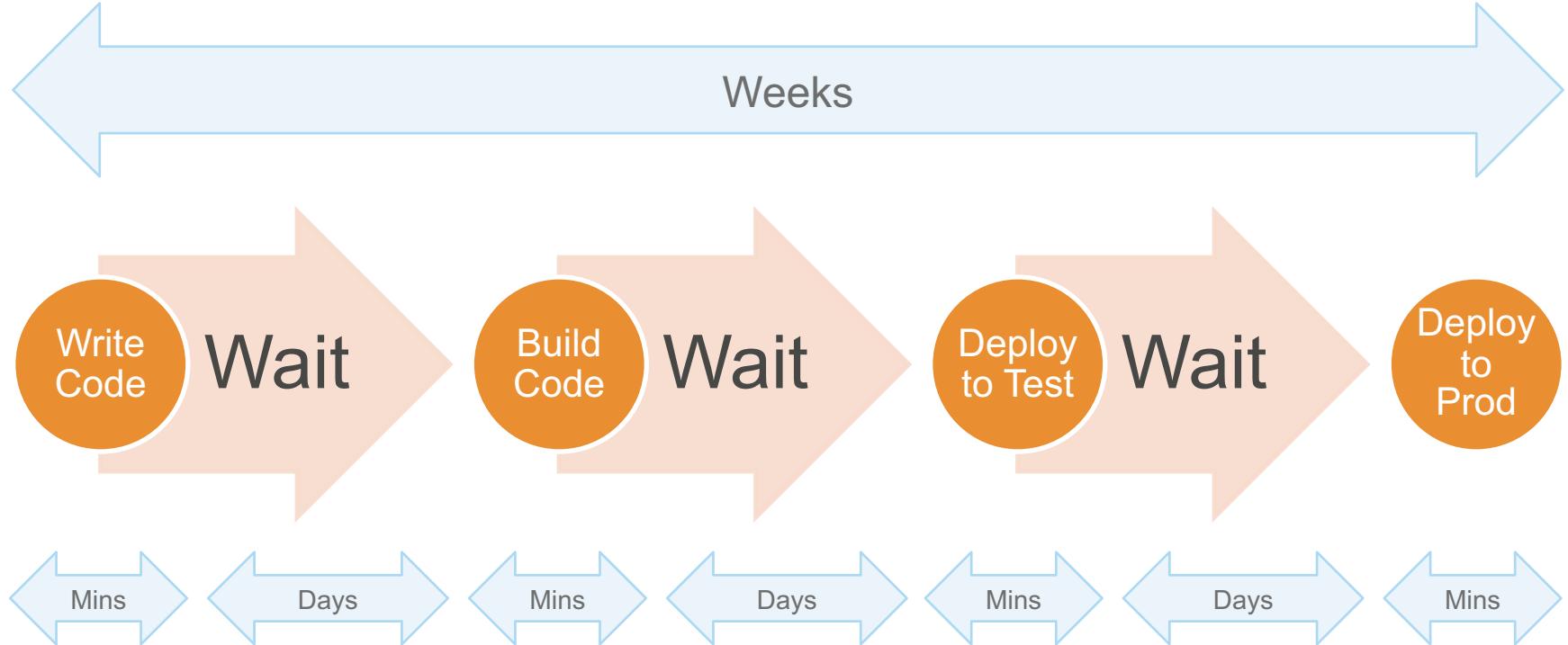
# We were just waiting.



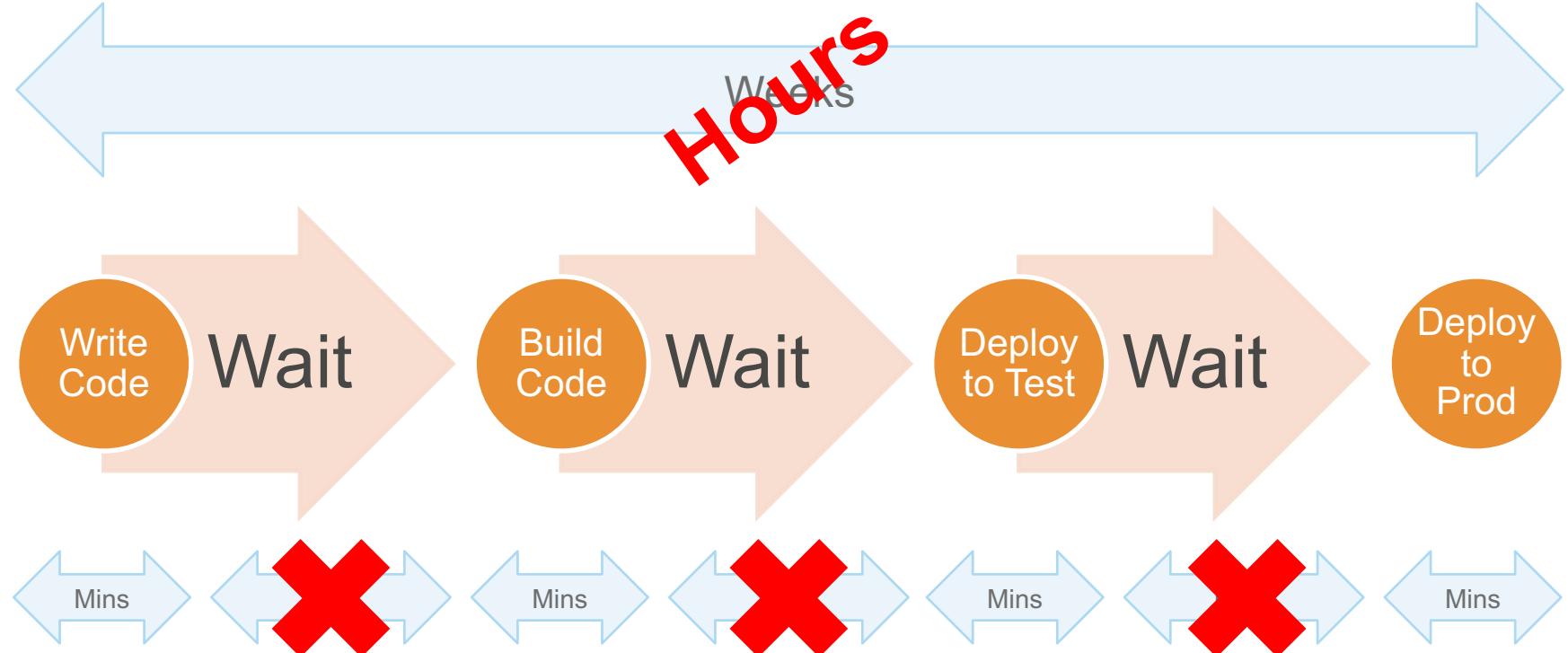
# We were just waiting.



# We were just waiting.



# We were just waiting.



A photograph showing a large stack of approximately 20-30 metal pipes, likely steel, lying horizontally on a bed of reddish-brown gravel. The pipes are weathered and show signs of rust and paint peeling. Some have handwritten markings like "12.12" and "12.55" near their ends. The lighting is bright, casting long shadows of the pipes onto the gravel.

We built tools to  
automate our software  
release process



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



# Pipelines

- Automated actions and transitions; from check-in to production
- Development benefits:
  - Faster
  - Safer
  - Simplification & standardization
  - Visualization of the process

# This has continued to work out really well:

- In 2014:
- Thousands of service teams across Amazon
- Building microservices
- Practicing continuous delivery
- Many environments (staging, beta, production)

- **50 million deploys**

# This has continued to work out really well:

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

**continuous delivery == happier developers!**

# Where do you

STAR?

# Introducing: AWS CodeStar

Quickly develop, build, and deploy applications on AWS



**Start developing on AWS in minutes**

**Work across your team, securely**

**Manage software delivery easily**

**Choose from a variety of project templates**

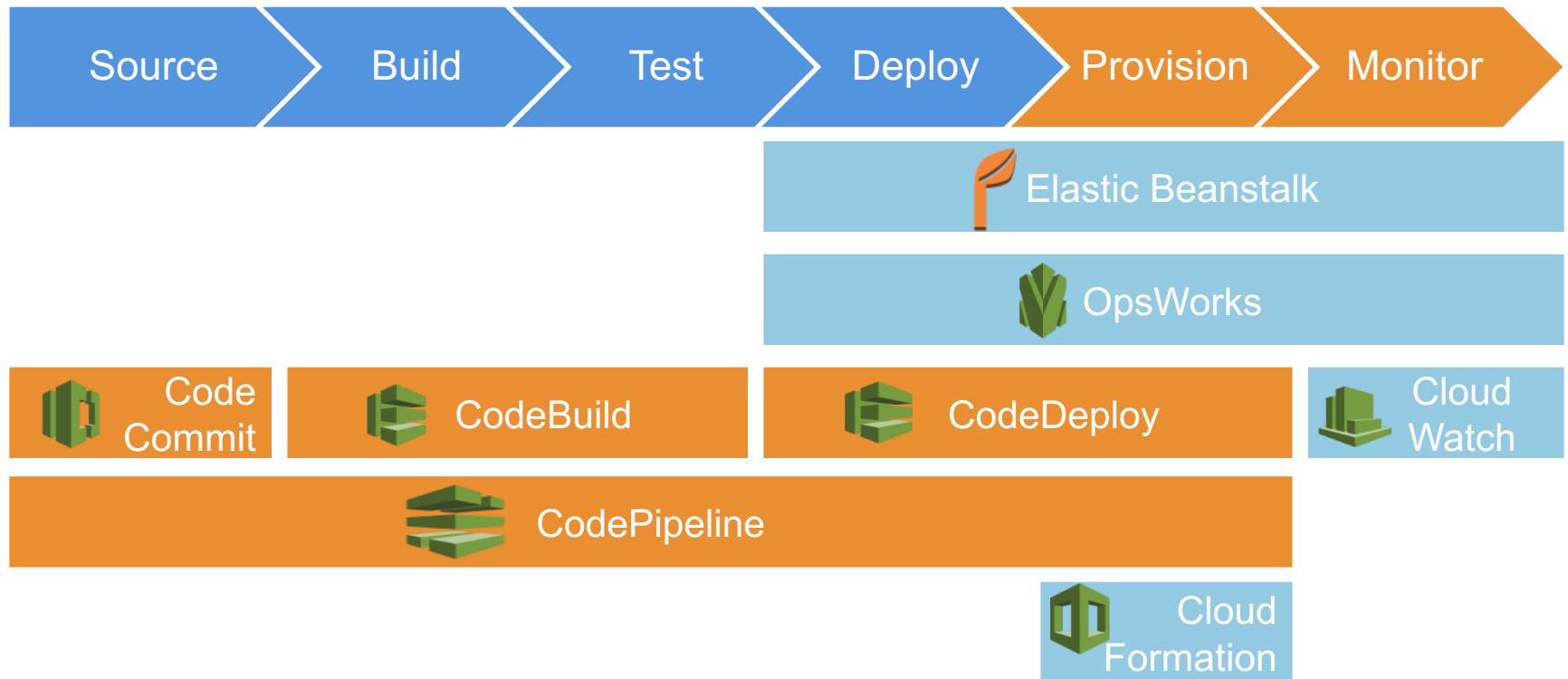
# Building a DevOps culture

## Enable DevOps change across:

- Culture
- Technology    ← Our Focus Today
- Management

Implement across all tracks in parallel, if possible

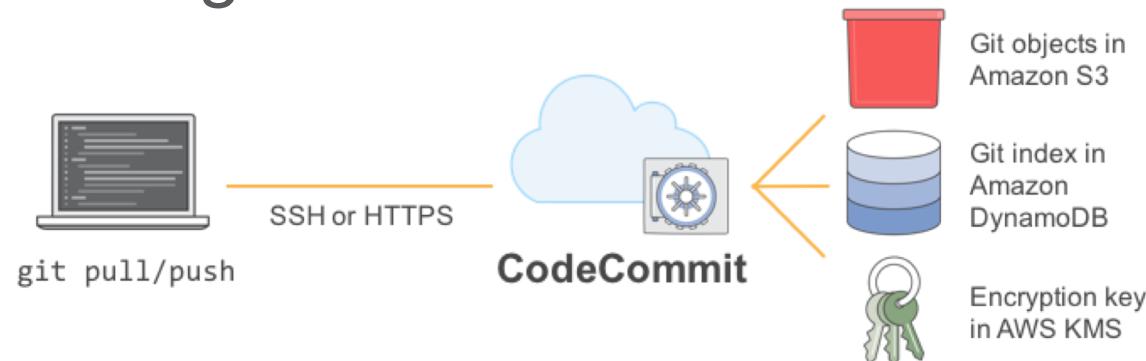
# Implementing DevOps on AWS



# AWS CodeCommit



- Secure, scalable, and managed Git source control
- Automatic encryption at rest and in transit
- Supports existing tooling
- Highly available



# AWS CodeBuild

- Fully managed build service with pay-as-you-go pricing
- Support for compiling source code, running tests, and building packages
- Enables continuous integration and delivery



# AWS CodeBuild Key Features

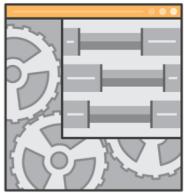
- Build & test code
  - Support for Java, Python, Node.js, Ruby, Go, Android, Docker, Windows (coming soon)
  - Install your own build tools and language runtimes
- Configurable build settings
  - Specify build commands
  - Three compute types based on memory and CPU
  - Source integrations (GitHub, AWS CodeCommit, Amazon S3)

# AWS CodePipeline

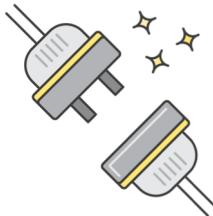


- Continuous delivery service for fast and reliable application updates
- Model and visualize your software release process
- Builds, tests, and deploys your code every time there is a code change
- Integrates with third-party tools and AWS

# AWS CodePipeline Benefits



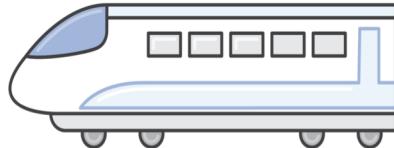
Configurable workflow



Easy to integrate



Improved quality



Rapid delivery

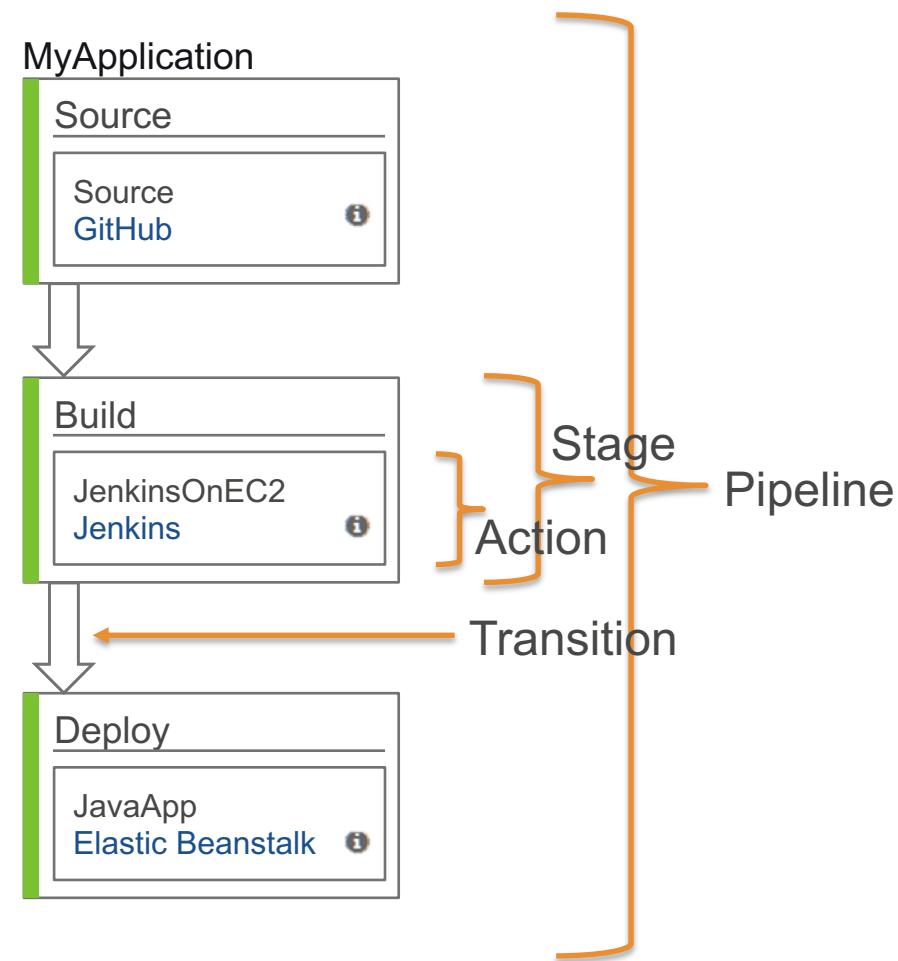


Get started fast

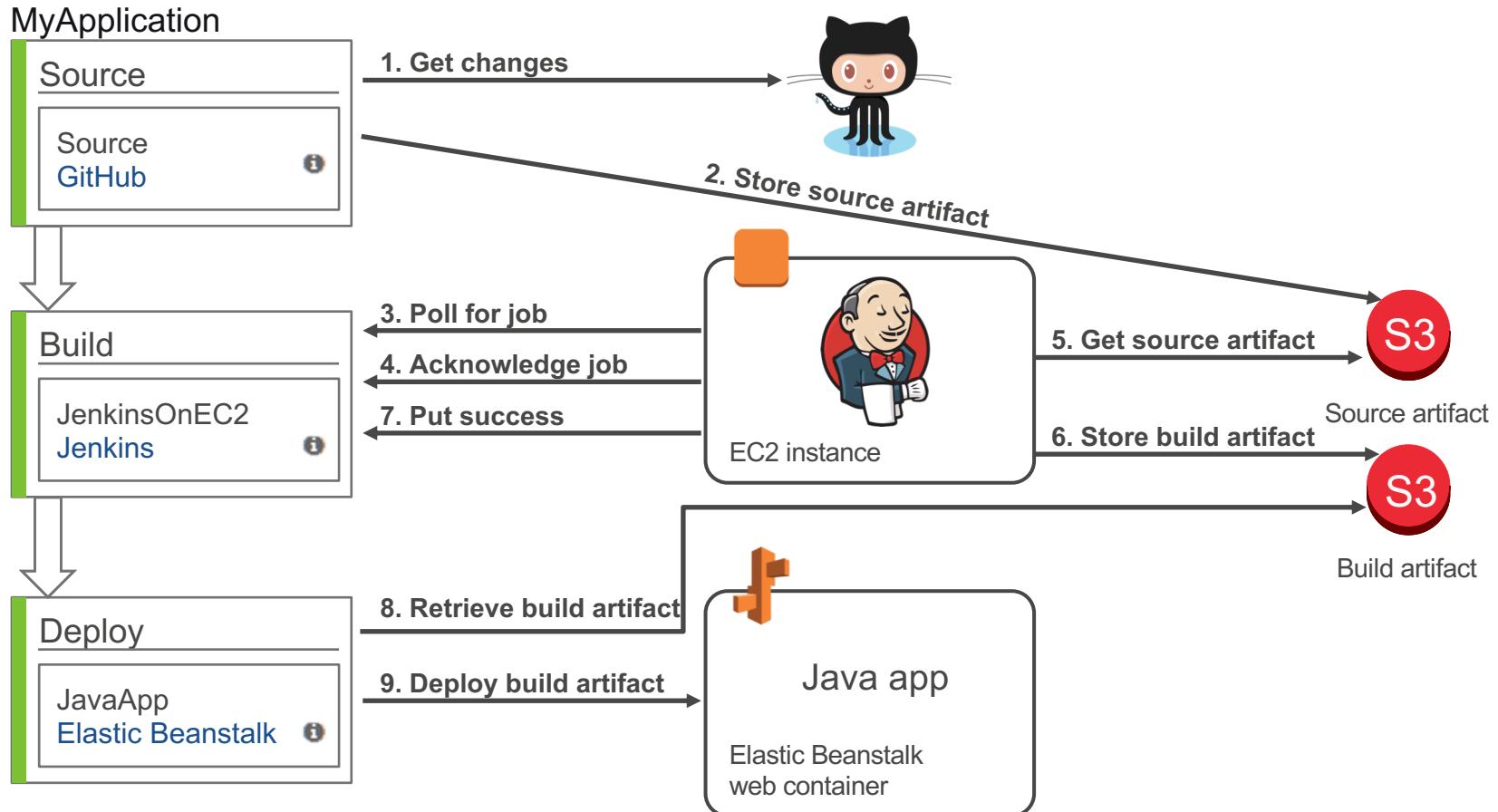
# Sample Pipeline

Includes support for:

- Parallel actions
- Sequential actions
- Manual approvals



# Sample Pipeline in practice



# Strong partners, and growing



**GitHub**



# AWS CodeDeploy



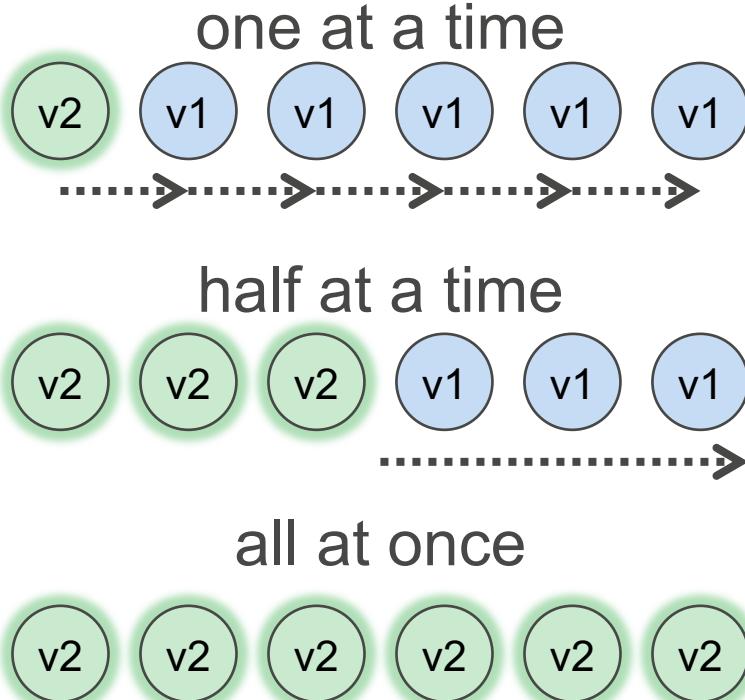
- Automates code deployments to any instance
- Handles the complexity of updating your applications
- Avoid downtime during application deployment
- Deploy to Amazon EC2 or on-premises servers, in any language and on any operating system
- Integrates with third-party tools and AWS

# CodeDeploy: appspec.yml

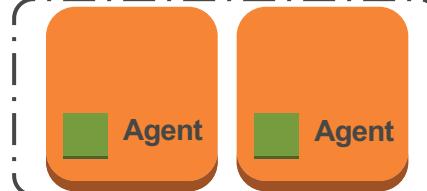
```
version: 0.0
os: linux
files:
  - source: /octank-ticketing.war
    destination: /tmp/codedeploy-deployment-staging-area/
  #- source: /scripts/configure_http_port.xsl
  #  destination: /tmp/codedeploy-deployment-staging-area/
hooks:
  ApplicationStop:
    - location: scripts/stop_application
      timeout: 300
  BeforeInstall:
    - location: scripts/install_dependencies
      timeout: 300
  ApplicationStart:
    - location: scripts/write_codedeploy_config.sh
    - location: scripts/start_application
      timeout: 300
  ValidateService:
    - location: scripts/basic_health_check.sh
```

- Send application files to one directory and configuration files to another
- Set specific permissions on specific directories and files
- Remove/add instance to ELB
- Install dependency packages
- Start Apache
- Confirm successful deploy
- More!

# Choose deployment speed and group



Dev deployment group



OR

Prod deployment group



# AWS CloudFormation



- Create templates of your infrastructure
- CloudFormation provisions AWS resources based on dependency needs
- Version control/replicate/update templates like code
- Integrates with development, CI/CD, management tools

# CloudFormation – Components & Technology

Template



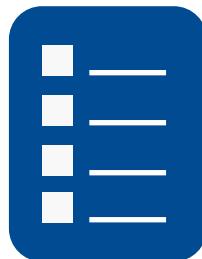
JSON/YAML formatted file

*Parameter definition*

*Resource creation*

*Configuration actions*

CloudFormation



Framework

*Stack creation*

*Stack updates*

*Error detection and rollback*

Stack



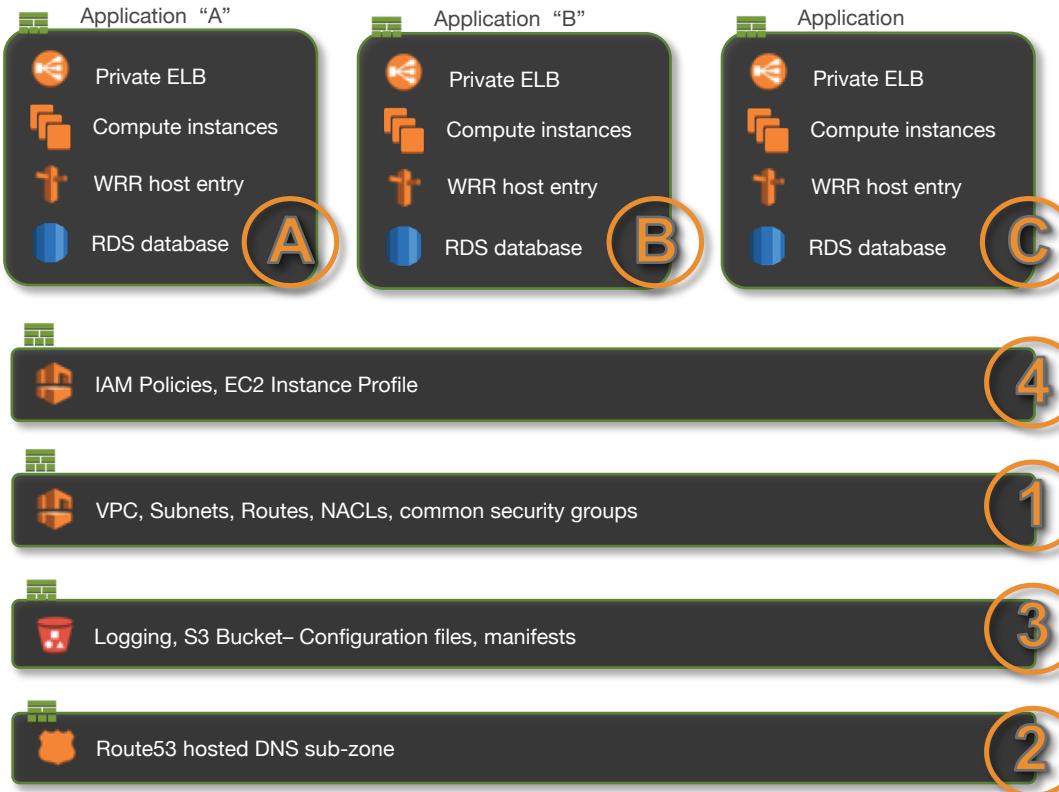
Configured AWS resources

*Comprehensive service support*

*Service event aware*

*Customizable*

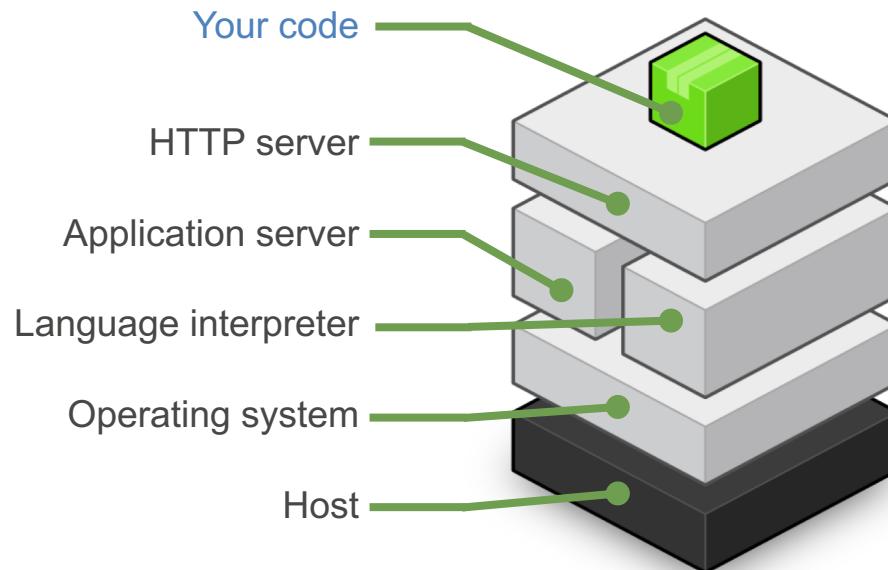
# CloudFormation stack layering



# Elastic Beanstalk



- Focus on building your application
- Web and worker tiers
- Provisions necessary infrastructure



# Benefits of Elastic Beanstalk

- Reduce time to market
- Enabler for continuous delivery
- Rolling or Blue/Green Deployments
- Lower IT burden



# AWS Container Services Landscape

## MANAGEMENT

Deployment, Scheduling,  
Scaling & Management of  
containerized applications



Amazon Elastic  
Container Service



Amazon Elastic  
Container Service  
for Kubernetes

## HOSTING

Where the containers run



Amazon EC2



AWS Fargate

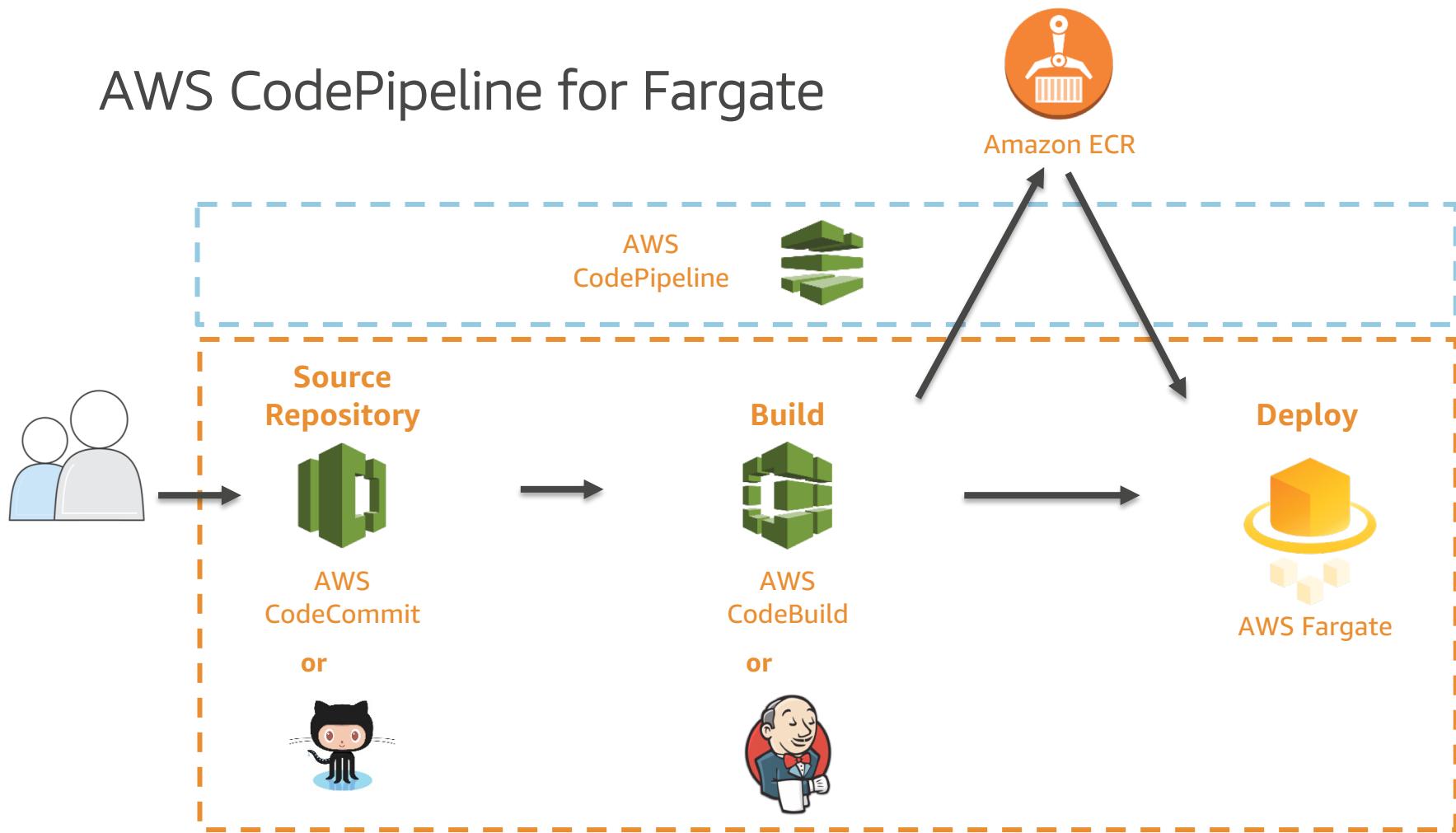
## IMAGE REGISTRY

Container Image Repository

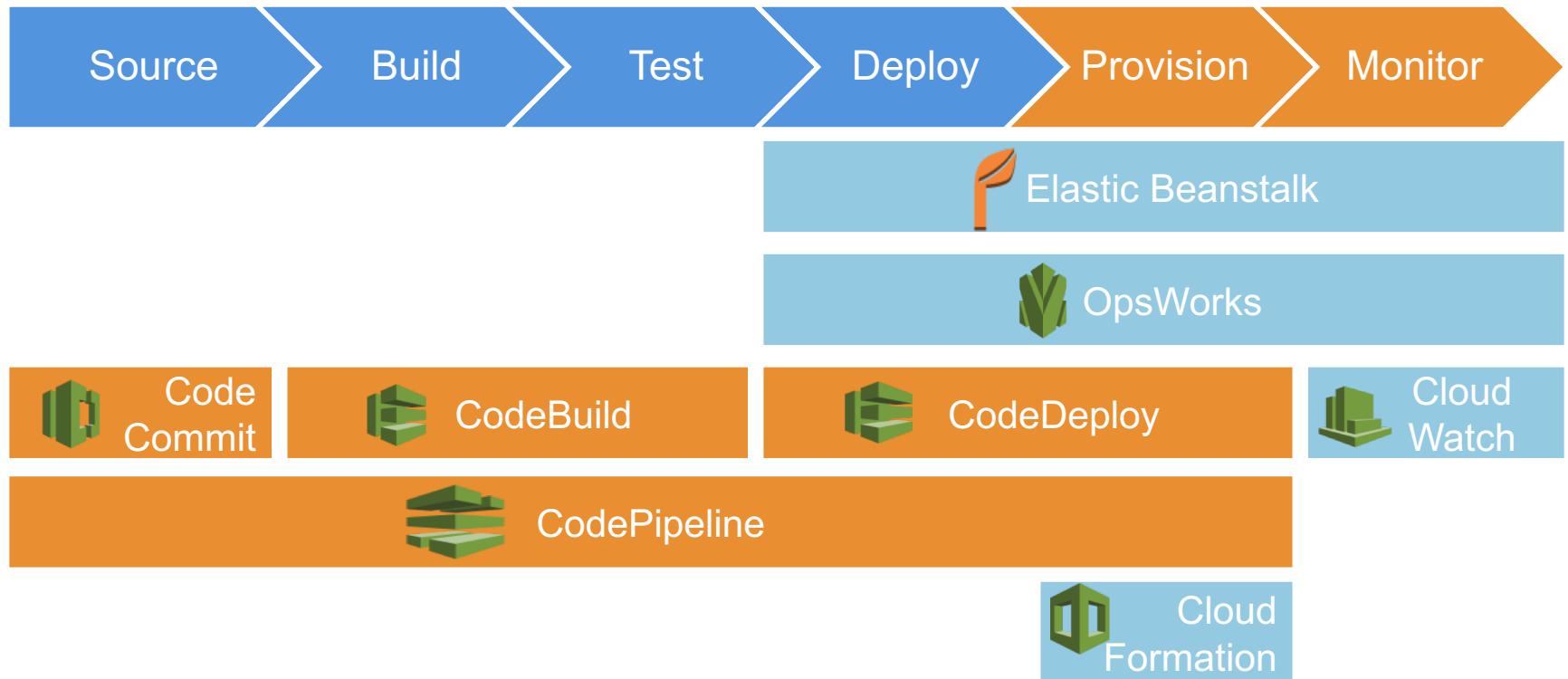


Amazon Elastic  
Container Registry

# AWS CodePipeline for Fargate



# Implementing DevOps on AWS



# Any Questions?



<https://aws.amazon.com/devops/>