Cloud Services 2

Amazon Web Services

PXL DIGITAL

AWS DevOps Tools

What is DevOps?

PXL DIGITAL

- Cultural philosophies
- Practices
- Tools

DevOps Culture

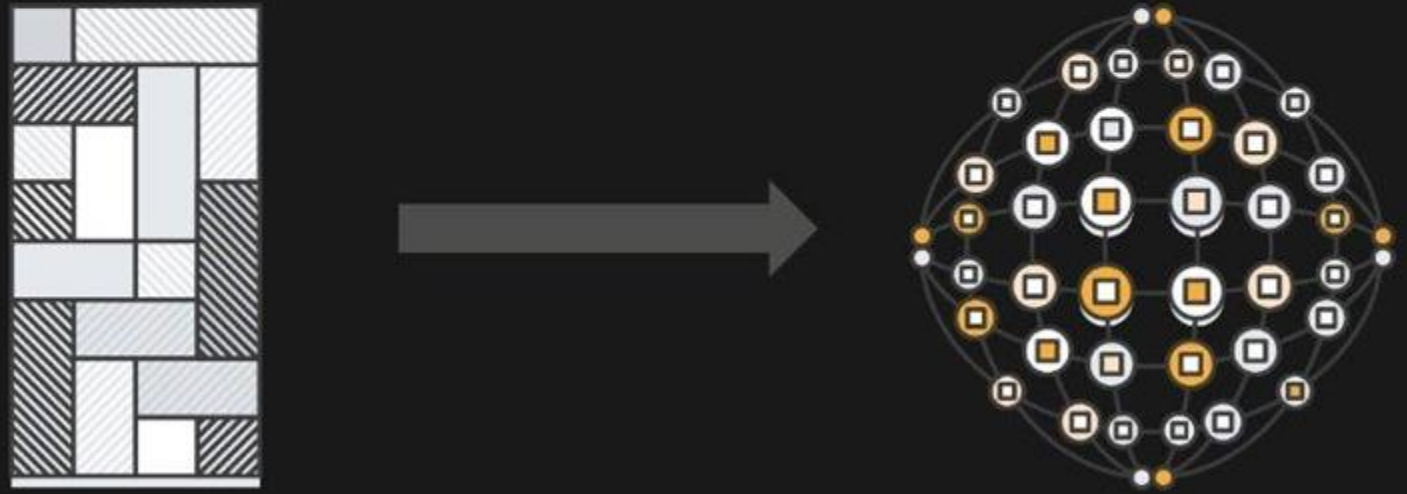
PXL DIGITAL

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

DevOps Practices

PXL DIGITAL

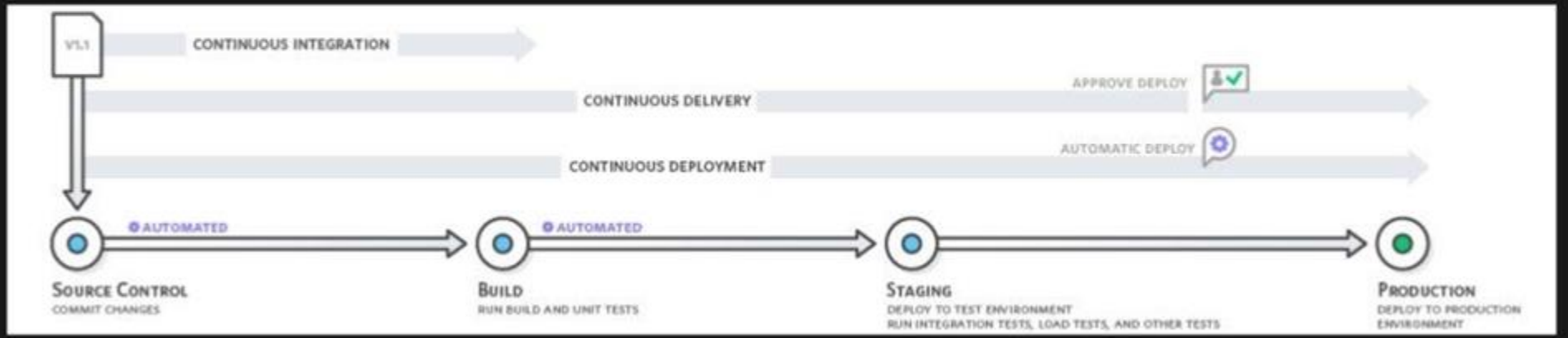
- Microservices
  - Moving away from “monolithic” application architecture to many individual services



DevOps Practices

PXL DIGITAL

- Continuous Integration
- Continuous Delivery & Deployment



DevOps Practices

PXL DIGITAL

- Infrastructure as Code
  - Model your AWS resources using code

DevOps Practices

- Monitoring and Logging
  - Track and analyze metrics and logs
  - Understand real-time performance of infrastructure and application

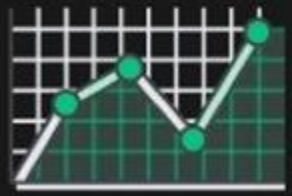
Benefits of DevOps

Improved Collaboration

Rapid Delivery

Reliability

Security



Scale

Speed

Five major phases of release and monitor

PXL DIGITAL

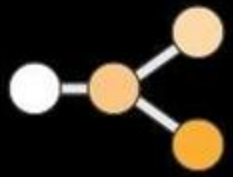
- Check-in source code such as .java files.
- Peer review new code

- Compile code
- Unit tests
- Style checkers
- Code metrics
- Create container images

- Integration tests with other systems
- Load testing
- UI tests
- Penetration testing

- Deployment to production environments

- Monitor code in production to quickly detect unusual activity or errors



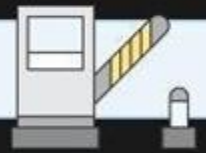
Release processes levels

PXL DIGITAL

Source → Build → Test → Production

Continuous integration

Continuous delivery



Continuous deployment

AWS Code Services

Software Release Steps:

PXL DIGITAL

AWS CodeStar

Source → Build → Test → Deploy → Monitor

AWS CodeCommit

AWS CodeBuild

AWS CodeBuild + Third Party

AWS CodeDeploy

AWS X-Ray

Amazon CloudWatch

AWS CodePipeline

AWS DevOps Portfolio

PXL DIGITAL

Software Development and Continuous Delivery Toolchain

AWS CodeStar

AWS CodeCommit

AWS CodeBuild

AWS CodeDeploy

AWS CodePipeline

Infrastructure as Code

AWS CloudFormation

AWS OpsWorks

AWS OpsWorks for Chef Automate

Monitoring & Logging

AWS X-Ray

Amazon CloudWatch

AWS CloudTrail

AWS Config


PXL DIGITAL
Build & test your application
https://secure.flickr.com/photos/spenceyc/7481166880

AWS CodeBuild

PXL DIGITAL

Fully managed build service that compiles source code, runs tests, and produces software packages

Scales continuously and processes multiple builds concurrently

You can provide custom build environments suited to your needs via Docker images

Only pay by the minute for the compute resources you use

Launched with CodePipeline and Jenkins integration

How can I automate my release process with CodeBuild?

PXL DIGITAL

- Integrated with AWS CodePipeline for CI/CD
- Easily pluggable (API/CLI driven)
- Bring your own build environments
    - Create Docker images containing tools you need
- Open source Jenkins plugin
    - Use CodeBuild as the workers off of a Jenkins master

buildspec.yml Example

```yaml
version: 0.1

environment_variables:
  plaintext:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"

phases:
  install:
    commands:
      - apt-get update -y
      - apt-get install -y maven
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  type: zip
  files:
    - target/messageUtil-1.0.jar
  discard-paths: yes
```

- Variables to be used by phases of build
- Examples for what you can do in the phases of a build:
  - You can install packages or run commands to prepare your environment in ”install”.
  - Run syntax checking, commands in “pre_build”.
  - Execute your build tool/command in “build”
  - Test your app further or ship a container image to a repository in post_build
- Create and store an artifact in S3

Building Your Code

“Building” code typically refers to languages that require compiled binaries:

- .NET languages: C#, F#, VB.net, etc.
- Java and JVM languages: Java, Scala, JRuby
- Go
- iOS languages: Swift, Objective-C

We also refer to the process of creating Docker container images as “building” the image.

Testing Your Code

PXL DIGITAL

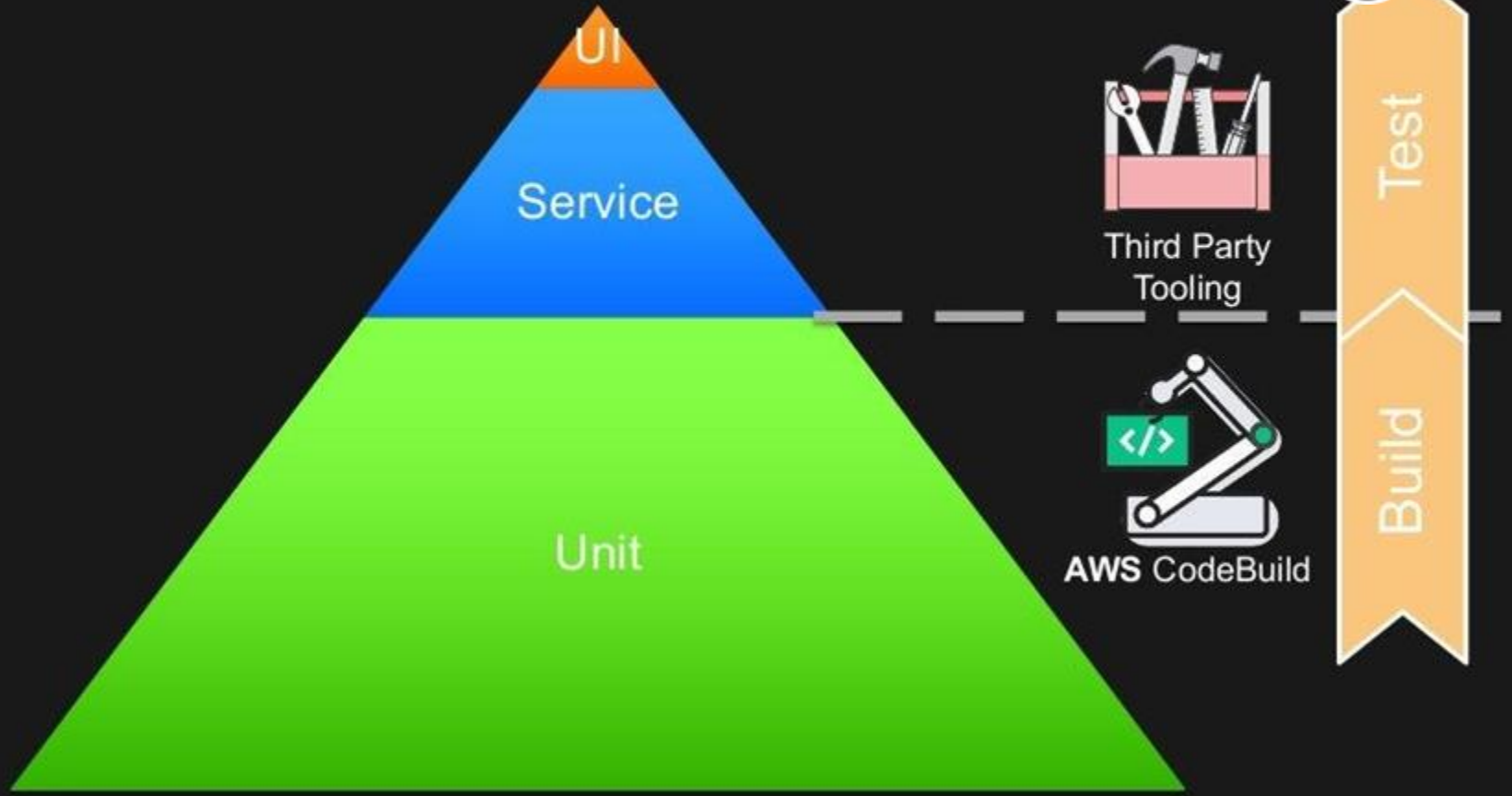
Testing is both a science and an art form!

Goals for testing your code:

- Want to confirm desired functionality
- Catch programming syntax errors
- Standardize code patterns and format
- Reduce bugs due to non-desired application usage and logic failures
- Make applications more secure

What service and release step corresponds with which tests?

PXL DIGITAL




PXL DIGITAL
Deploying your applications
https://secure.flickr.com/photos/simononly/15386966677

AWS CodeDeploy

Automates code deployments to any instance

Handles the complexity of updating your applications

Avoid downtime during application deployment

Rollback automatically if failure detected

Deploy to Amazon EC2 or on-premises servers, in any language and on any operating system

Integrates with third-party tools and AWS

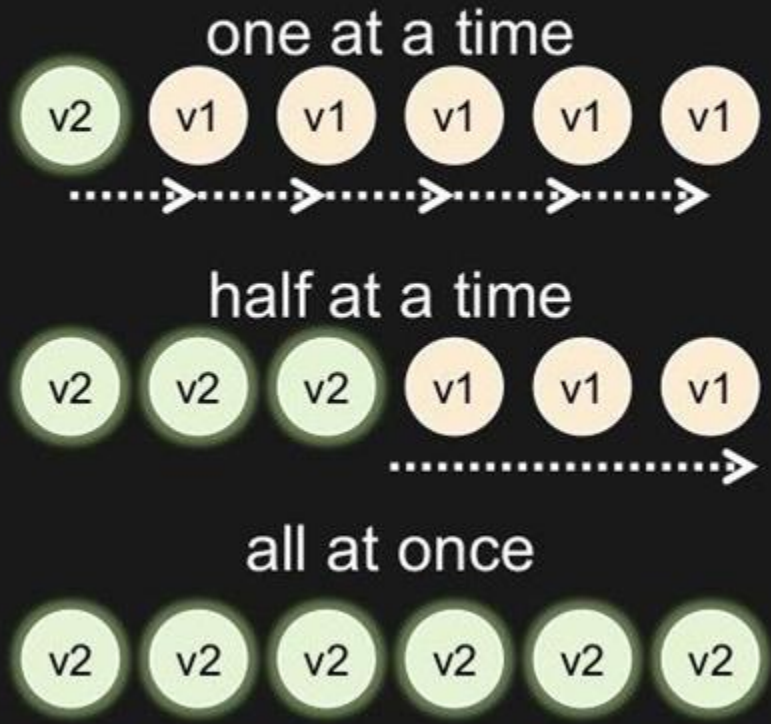
appspec.yml Example

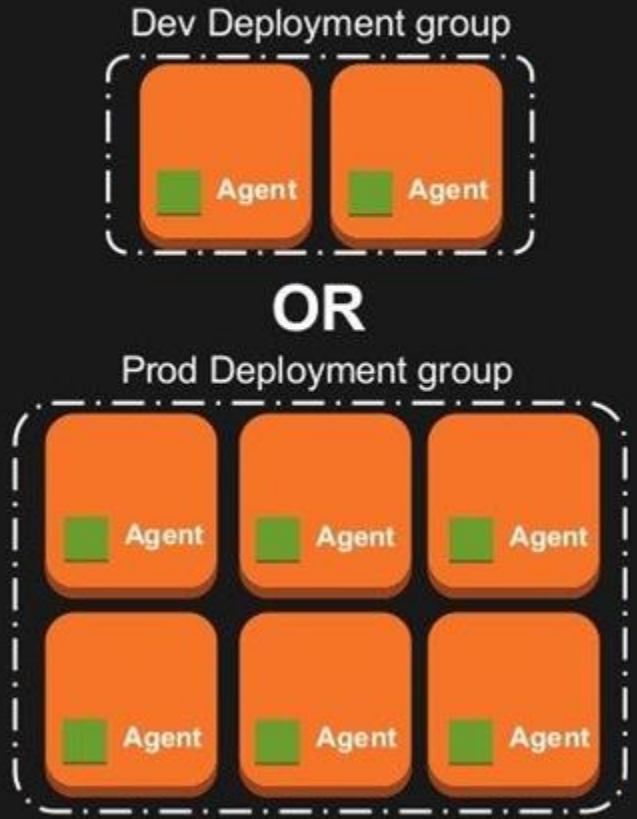
```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
permissions:
  - object: /var/www/html
    pattern: “*.html”
    owner: root
    group: root
    mode: 755
hooks:
  ApplicationStop:
    - location: scripts/deregister_from_elb.sh
  BeforeInstall:
    - location: scripts/install_dependencies.sh
  ApplicationStart:
    - location: scripts/start_httpd.sh
  ValidateService:
    - location: scripts/test_site.sh
    - location: scripts/register_with_elb.sh
```

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

Choose Deployment Speed and Group

PXL DIGITAL

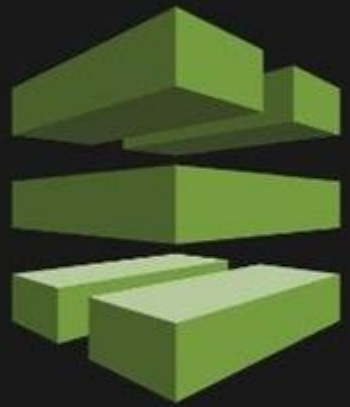





PXL DIGITAL
Orchestrating build and deploy with a pipeline
https://www.flickr.com/photos/seattlemunicipalarchives/12504672623/

AWS CodePipeline

PXL DIGITAL

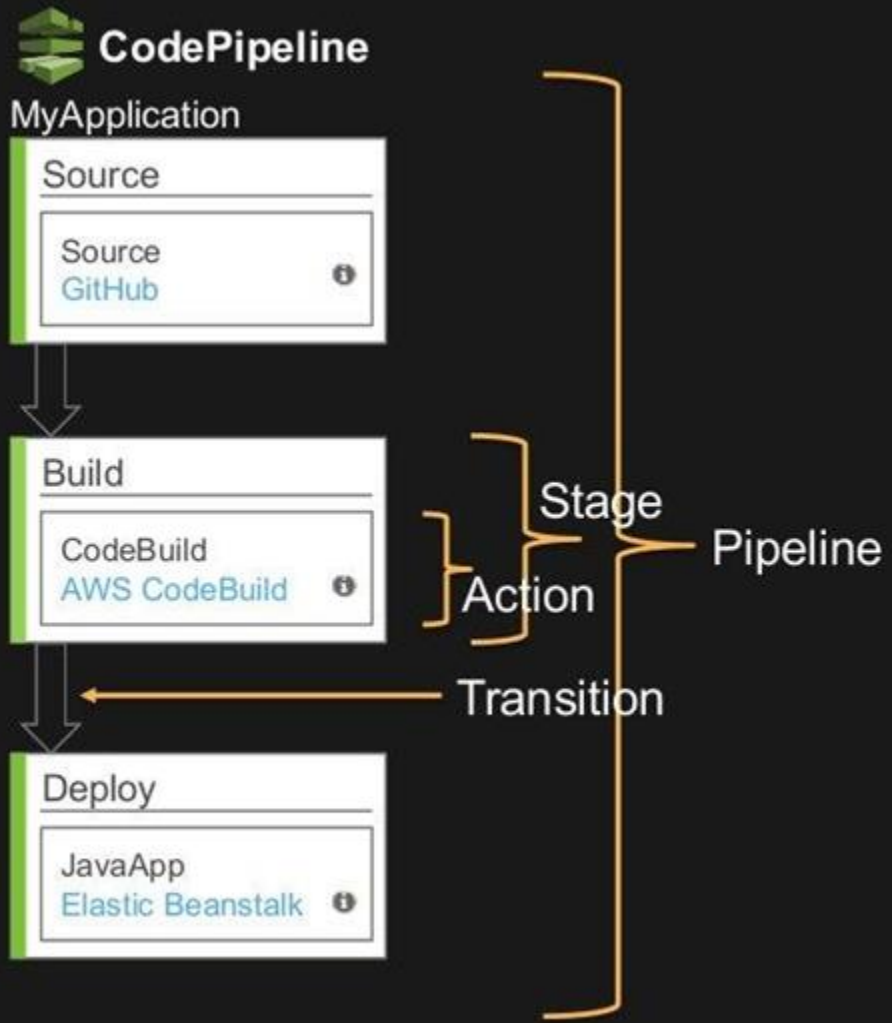


Continuous delivery service for fast and reliable application updates

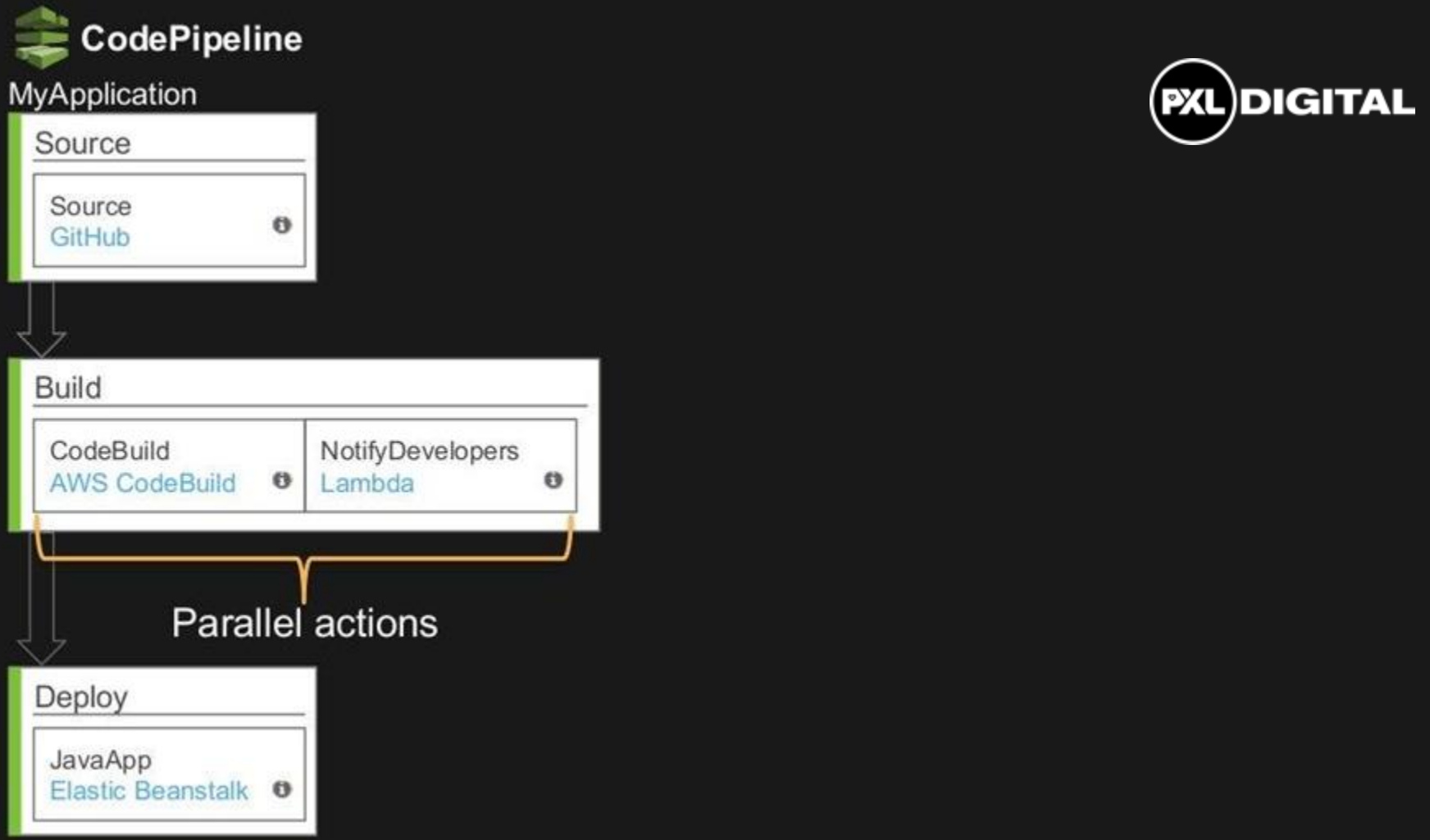
Model and visualize your software release process

Builds, tests, and deploys your code every time there is a code change

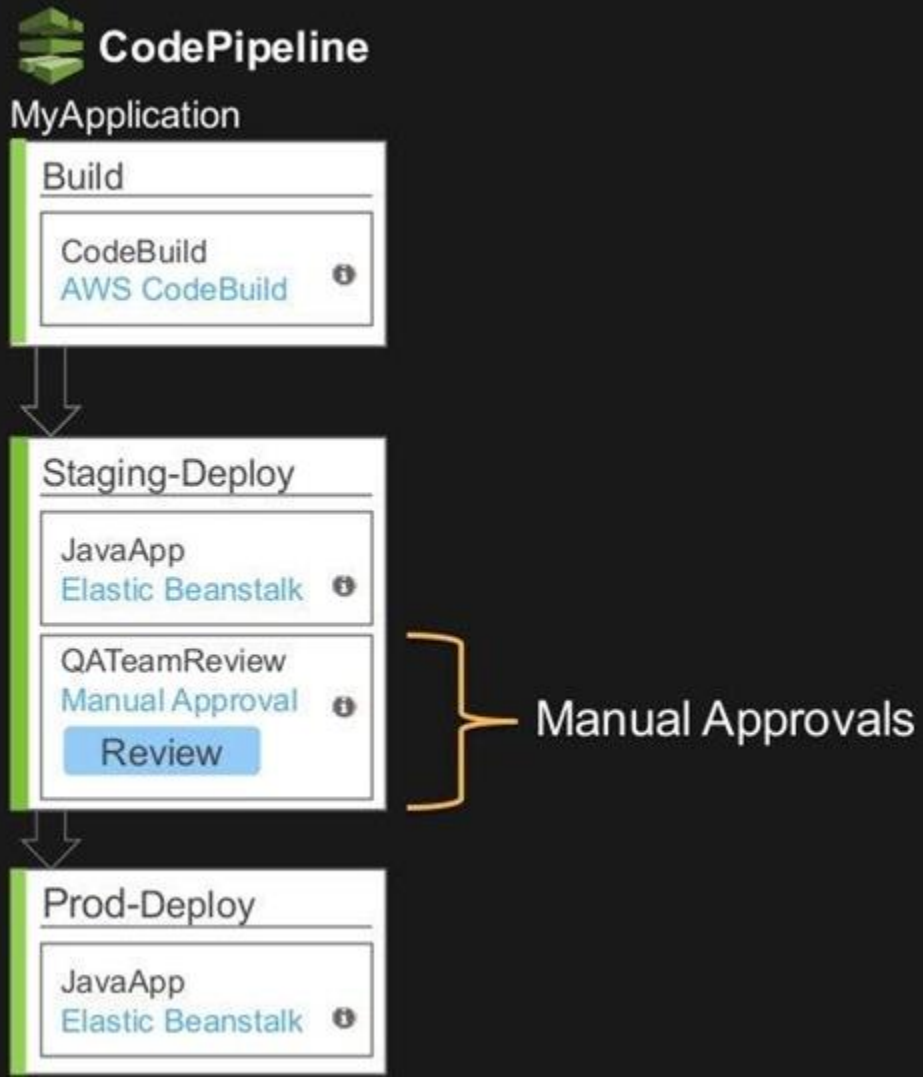
Integrates with third-party tools and AWS


CodePipeline
MyApplication
Source
Source
GitHub
Build
CodeBuild
AWS CodeBuild
Deploy
JavaApp
Elastic Beanstalk
Stage
Action
Pipeline
Transition

PXL DIGITAL


CodePipeline
MyApplication
Source
Source
GitHub
Build
CodeBuild
AWS CodeBuild
NotifyDevelopers
Lambda
Parallel actions
Deploy
JavaApp
Elastic Beanstalk
PXL DIGITAL


CodePipeline
MyApplication
Source
Source
GitHub
Build
CodeBuild
AWS CodeBuild
NotifyDevelopers
Lambda
TestAPI
Runscope
Sequential actions
Deploy
JavaApp
Elastic Beanstalk
PXL DIGITAL


CodePipeline
MyApplication
Build
CodeBuild
AWS CodeBuild
Staging-Deploy
JavaApp
Elastic Beanstalk
QATeamReview
Manual Approval
Review
Manual Approvals
Prod-Deploy
JavaApp
Elastic Beanstalk

AWS CodeCommit

Secure, scalable, and managed Git source control

Use standard Git tools

Scalability, availability, and durability of Amazon S3

Encryption at rest with customer-specific keys

No repo size limit

Post commit hooks to call out to SNS/Lambda

AWS CodeCommit

PXL DIGITAL

AWS X-Ray

Debug and analyze production applications in cloud or on-prem

Visualize service graph to identify performance bottlenecks

Troubleshoot and fix performance issues

Quantify Customer Impact

Integration with Lambda allows you to monitor Serverless applications

X-Ray SDK available in Java, .NET, Node.js, and Python

DevOps Engineering Best Practices

- Develop on main line
- Only Build Your Binaries Once
- Deploy the Same Way to Every Environment
- Smoke-Test Your Deployments
- Deploy into a Copy of Production
- Each Change Should Propagate through the Pipeline Instantly
- If Any Part of the Pipeline Fails, Stop the Line

Extra leermaterialen & labs

Bronvermelding slides:

- AWS Techtalk: https://www.slideshare.net/AmazonWebServices/devops-on-aws-devops-day-san-francisco