



AWS
SUMMIT

A Tale of Two Pizzas

Accelerating Software Delivery with AWS Developer Tools

9/12/18



A photograph of a weathered brick wall. A large, white, sans-serif question mark is superimposed on the wall, centered horizontally and vertically. The wall shows signs of age, including discoloration and small metal plates or bolts. In the top left corner, a portion of a building with a balcony and laundry hanging outside is visible.

Why are we
here today?

What we'll cover

What is DevOps?

The Amazon DevOps story

AWS Code Services

AWS DevOps Portfolio



Software moves
faster today

87

Why does DevOps matter?

5x

Lower change
failure rate

440x

Faster from commit
to deploy

46x

More frequent
deployments

44%

More time spent on
new features and
code

What is DevOps?

What is DevOps?

- Cultural philosophies
- Practices
- Tools

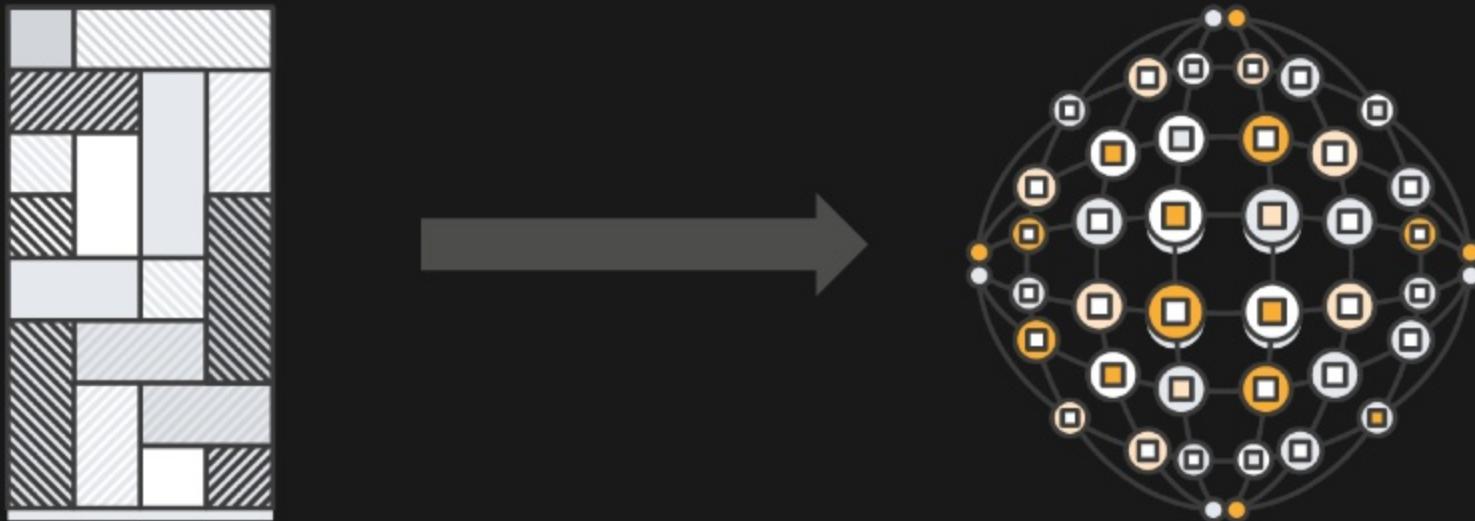
DevOps Culture

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



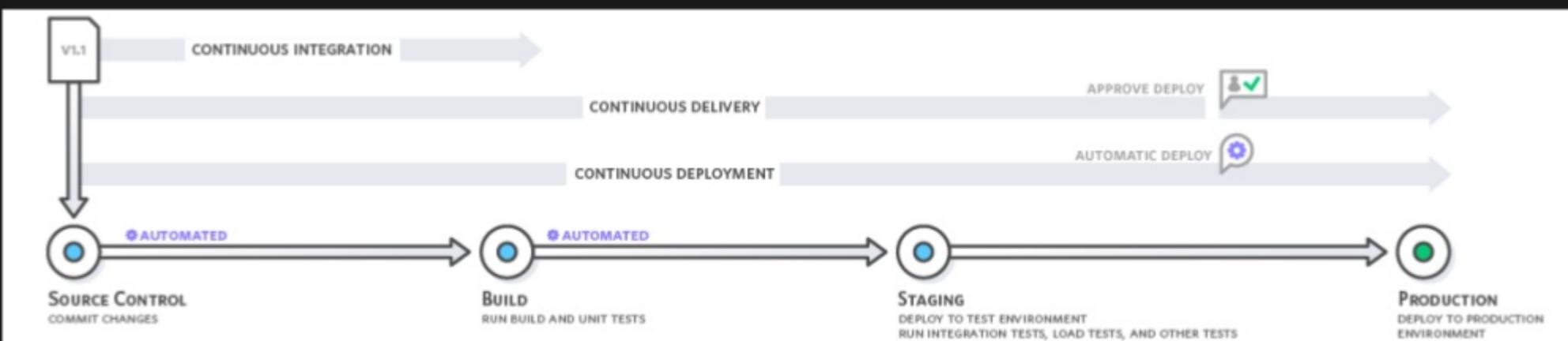
DevOps Practices

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



DevOps Practices

- Continuous Integration
- Continuous Delivery & Deployment



DevOps Practices

- Infrastructure as Code
 - Model your AWS resources using code

The screenshot shows the AWS CloudFormation console interface. At the top, there are tabs for Parameters, Mappings, Conditions, Metadata, and Outputs. The Parameters tab is selected. Below the tabs, the template name is "template1" with an edit icon. The main area displays the AWS CloudFormation JSON template code.

```
1+ {
2+   "Parameters": {
3+     "KeyPairName": {
4+       "Description": "Public/private key pairs allow you to securely connect to your instance after it launches",
5+       "Type": "AWS::EC2::KeyPair::KeyName"
6+     },
7+     "ADInstanceType": {
8+       "Description": "Amazon EC2 instance type for the first Active Directory Instance",
9+       "Type": "String",
10+      "Default": "m4.xlarge",
11+      "AllowedValues": [
12+        "m4.large",
13+        "m4.xlarge",
14+        "m4.2xlarge",
15+        "m4.4xlarge"
16+      ],
17+    },
18+    "AD2InstanceType": {
19+      "Description": "Amazon EC2 instance type for the second Active Directory Instance",
20+      "Type": "String",
21+      "Default": "m4.xlarge",
22+      "AllowedValues": [
23+        "m4.large",
24+        "m4.xlarge",
25+        "m4.2xlarge",
26+        "m4.4xlarge"
27+      ]
28+    }
29+  }
30+}
```

DevOps Practices

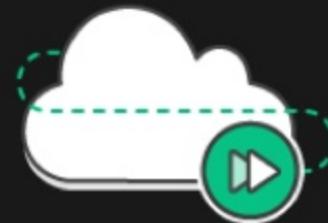
- Monitoring and Logging
 - Track and analyze metrics and logs
 - Understand real-time performance of infrastructure and application



Benefits of DevOps



Improved Collaboration



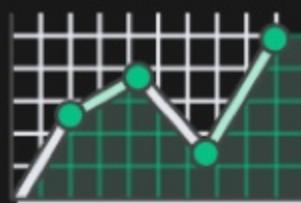
Rapid Delivery



Reliability



Security



Scale



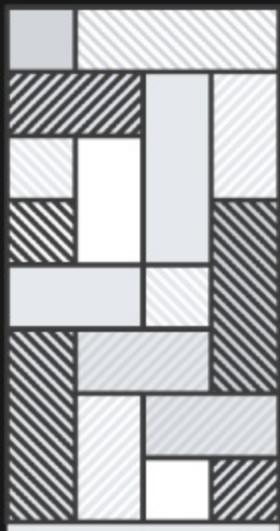
Speed

The background image shows a grand, multi-story stone staircase with intricate railings and a series of arched windows above. The architecture is highly detailed, suggesting a historical or classical setting like a museum or cathedral. Several people are visible on the stairs and landings, providing a sense of scale to the massive structure.

A look back at
development at
Amazon...

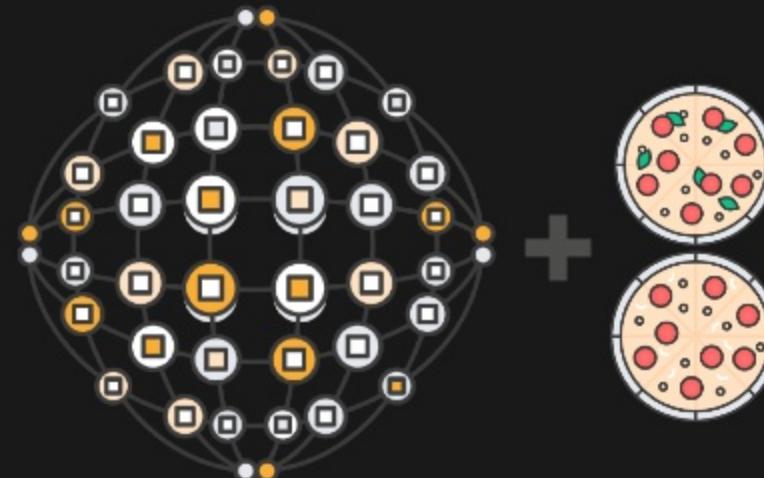
Development transformation at Amazon: 2001-2009

2001



monolithic
application + teams

2009



microservices + 2 pizza teams

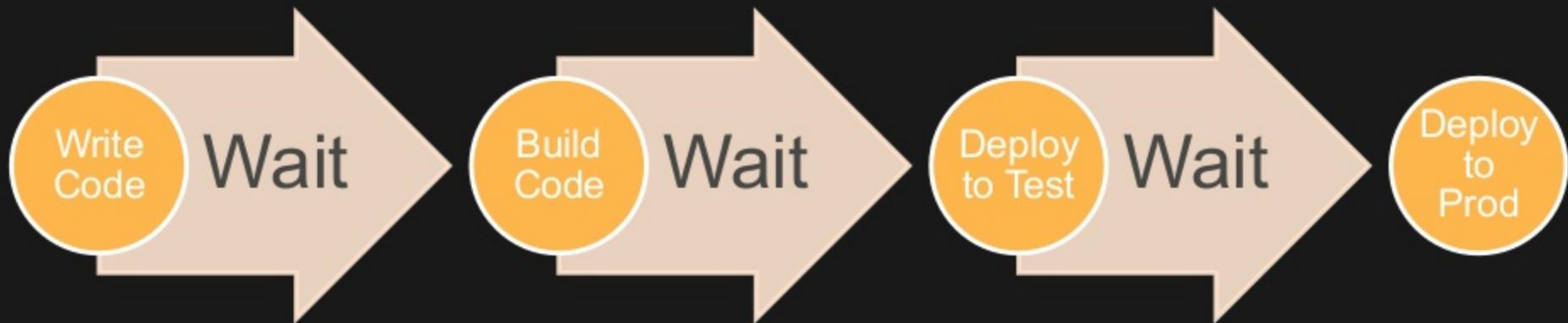
Things went much better under this model and teams were releasing faster than ever, but we felt that we could still improve.



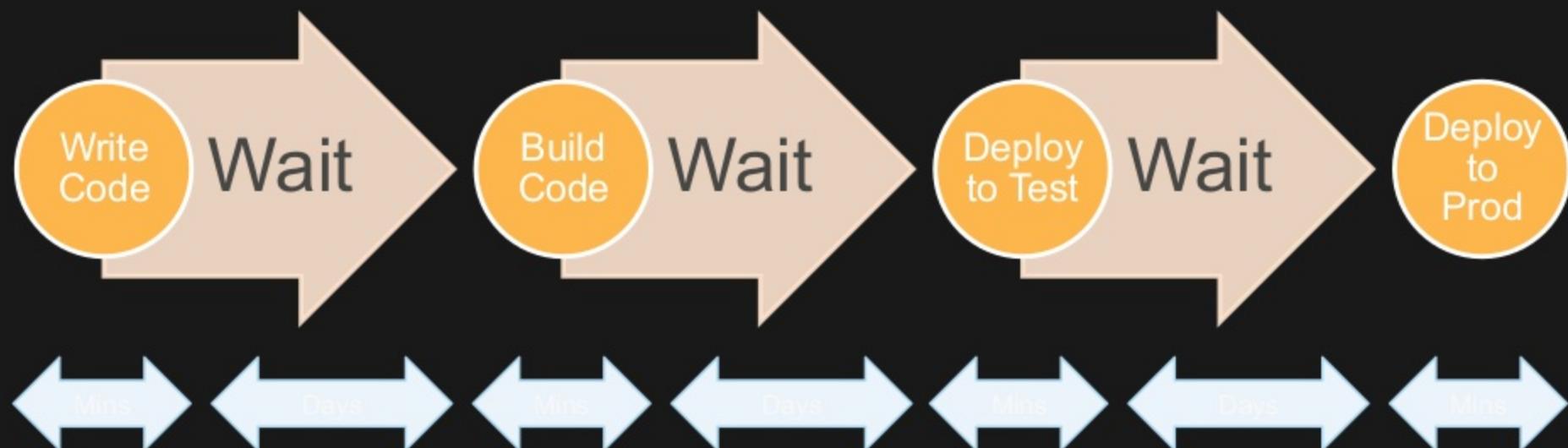


In 2009, we ran a study to find out where inefficiencies might still exist

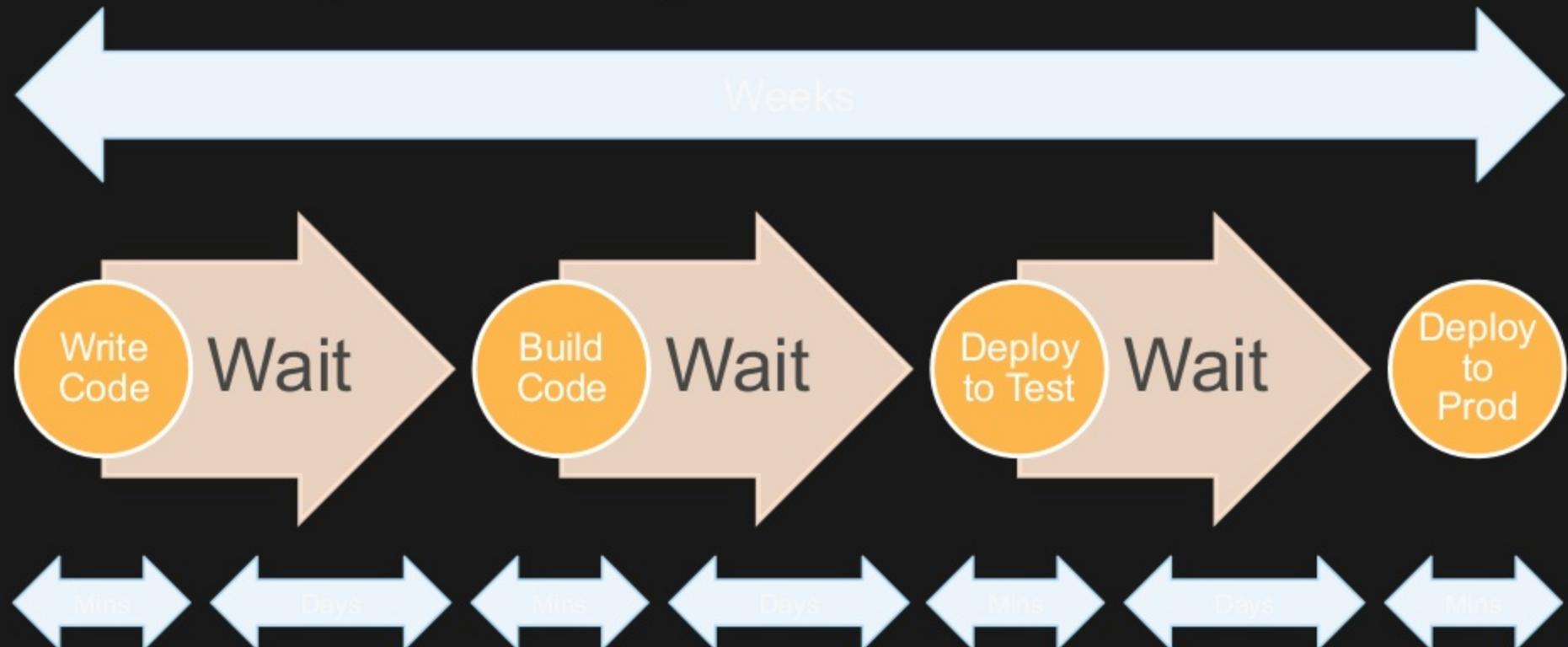
We were just waiting.



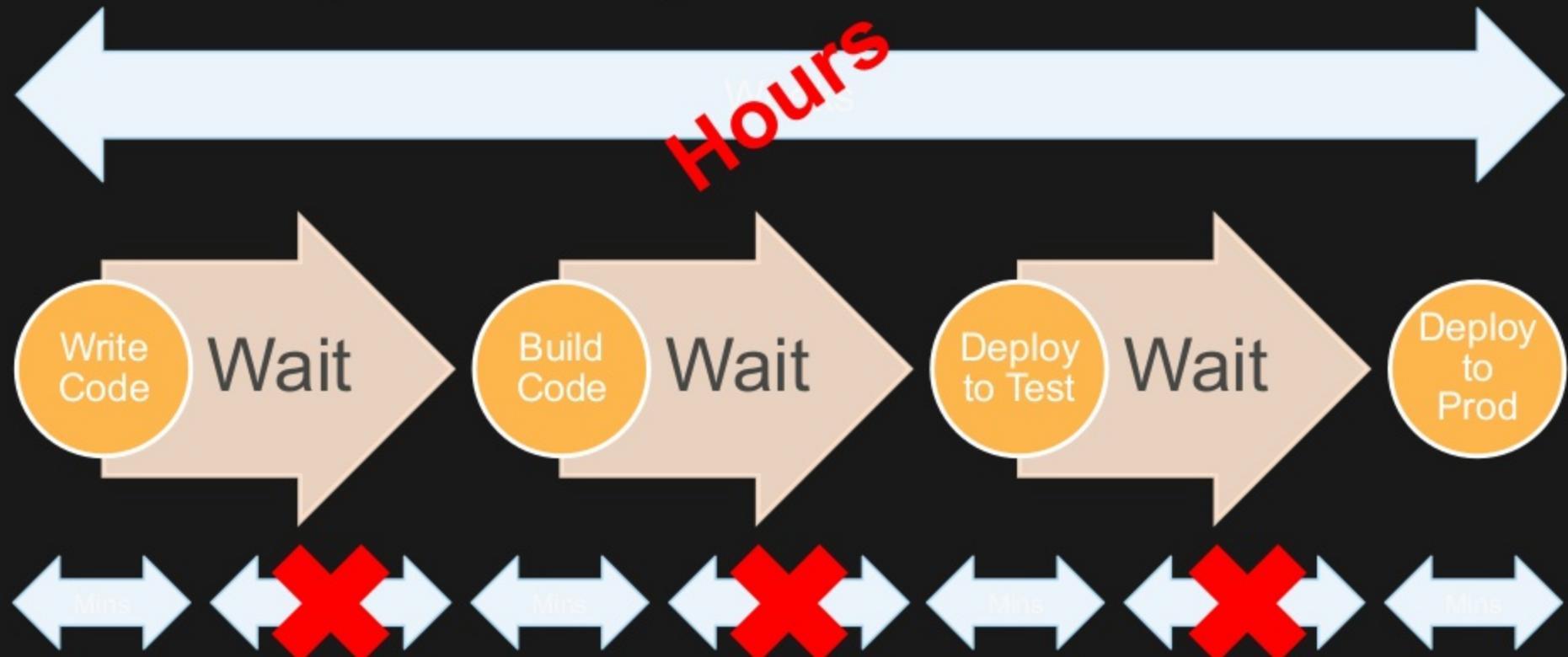
We were just waiting.



We were just waiting.



We were just waiting.



A photograph showing a stack of several blue and yellow painted metal pipes, likely steel, resting on a dark, gravelly ground. The pipes are arranged in a roughly triangular stack, with their ends pointing towards the bottom center of the frame. The paint on the pipes is worn and peeling, especially at the joints and ends. Some handwritten markings are visible on the pipes, such as "12.12" and "12.55". The lighting is bright, casting long shadows of the pipes onto the gravel.

We built tools to
automate our software
release process



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 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!

Where do you

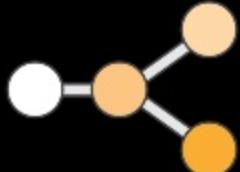
STAR?

?

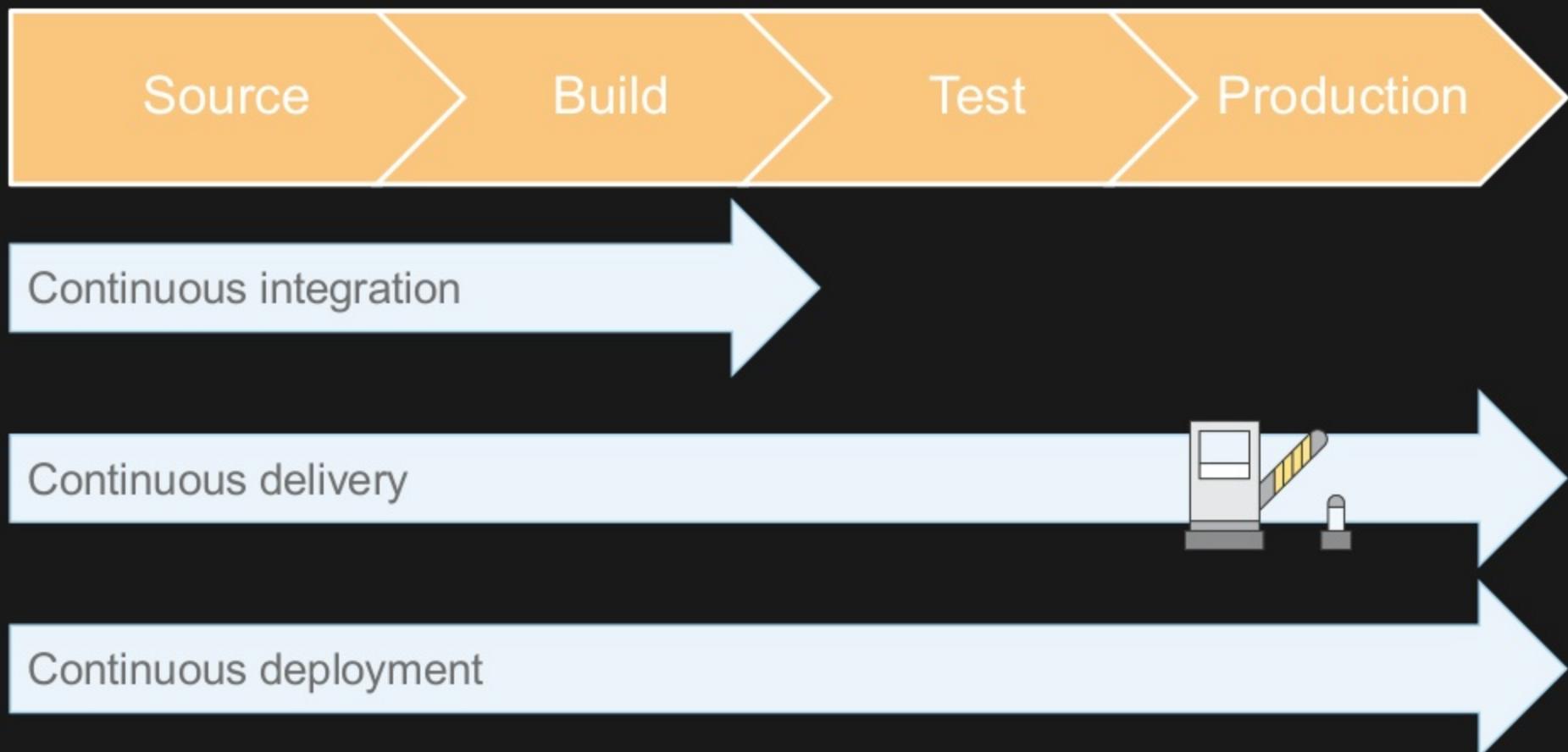
Five major phases of release and monitor



- 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

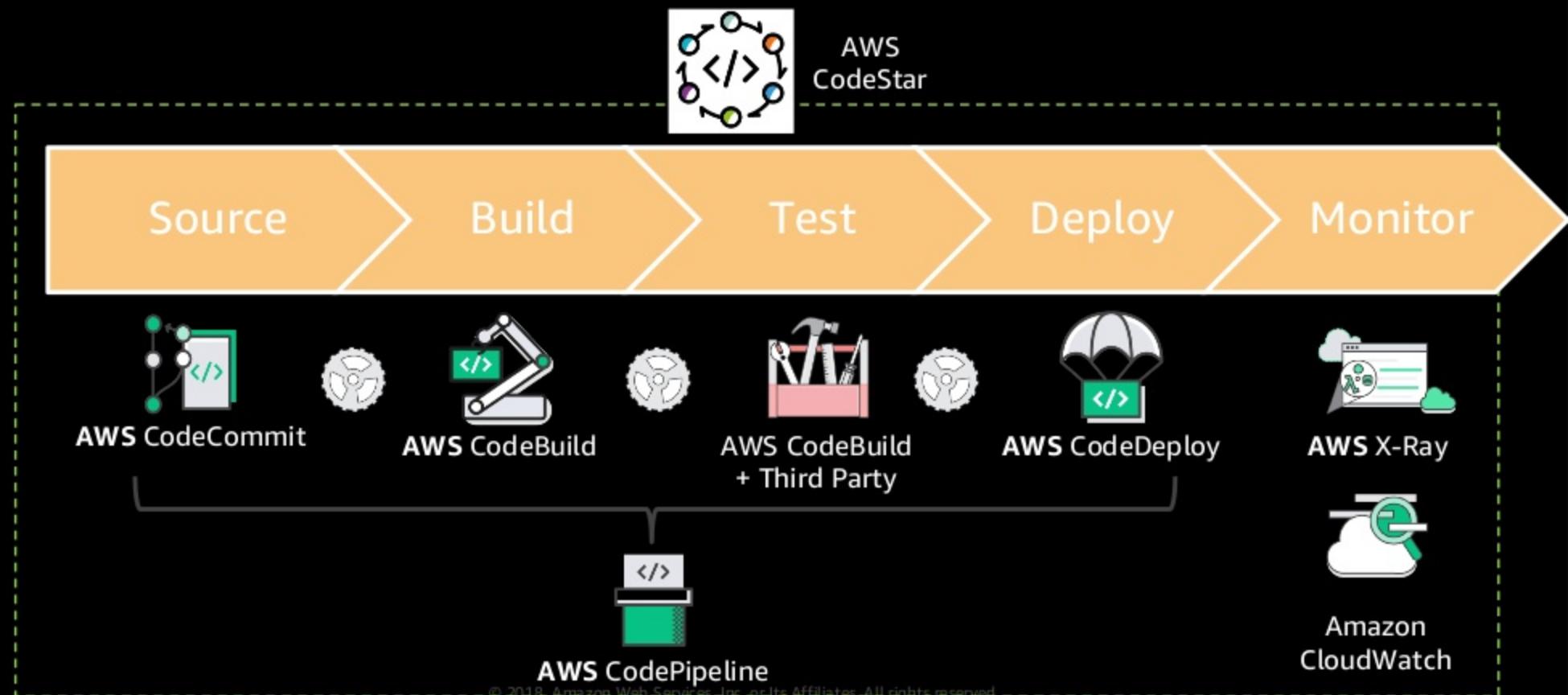


Release processes levels



AWS Code Services

Software Release Steps:



AWS DevOps Portfolio

Software Development and
Continuous Delivery Toolchain



AWS CodeCommit



AWS CodeStar



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

A photograph of an industrial assembly line, likely for car manufacturing. Numerous bright orange KUKA brand industrial robots are positioned along a conveyor belt, each working on different parts of a vehicle chassis. The robots are mounted on a complex system of overhead tracks and support structures. In the background, more robotic arms and factory equipment are visible under a high ceiling.

Build & test your
application

AWS CodeBuild

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 does it work?

1. Downloads source code
2. Executes commands configured in the buildspec in temporary compute containers (created fresh on every build)
3. Streams the build output to the service console and CloudWatch logs
4. Uploads the generated artifact to an S3 bucket

How can I automate my release process with CodeBuild?

- 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

```
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
```

buildspec.yml Example

```
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

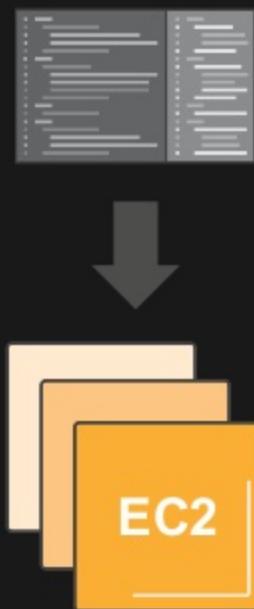
We also refer to the process of creating Docker container images as “building” the image.



No Building Required!

Many languages don't require building. These are considered interpreted languages:

- PHP
- Ruby
- Python
- Node.js



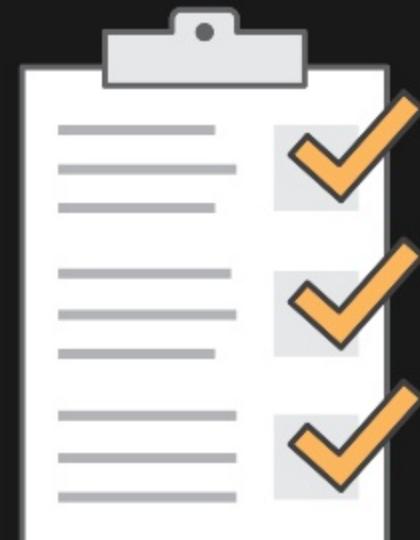
You can just deploy your code!

Testing Your Code

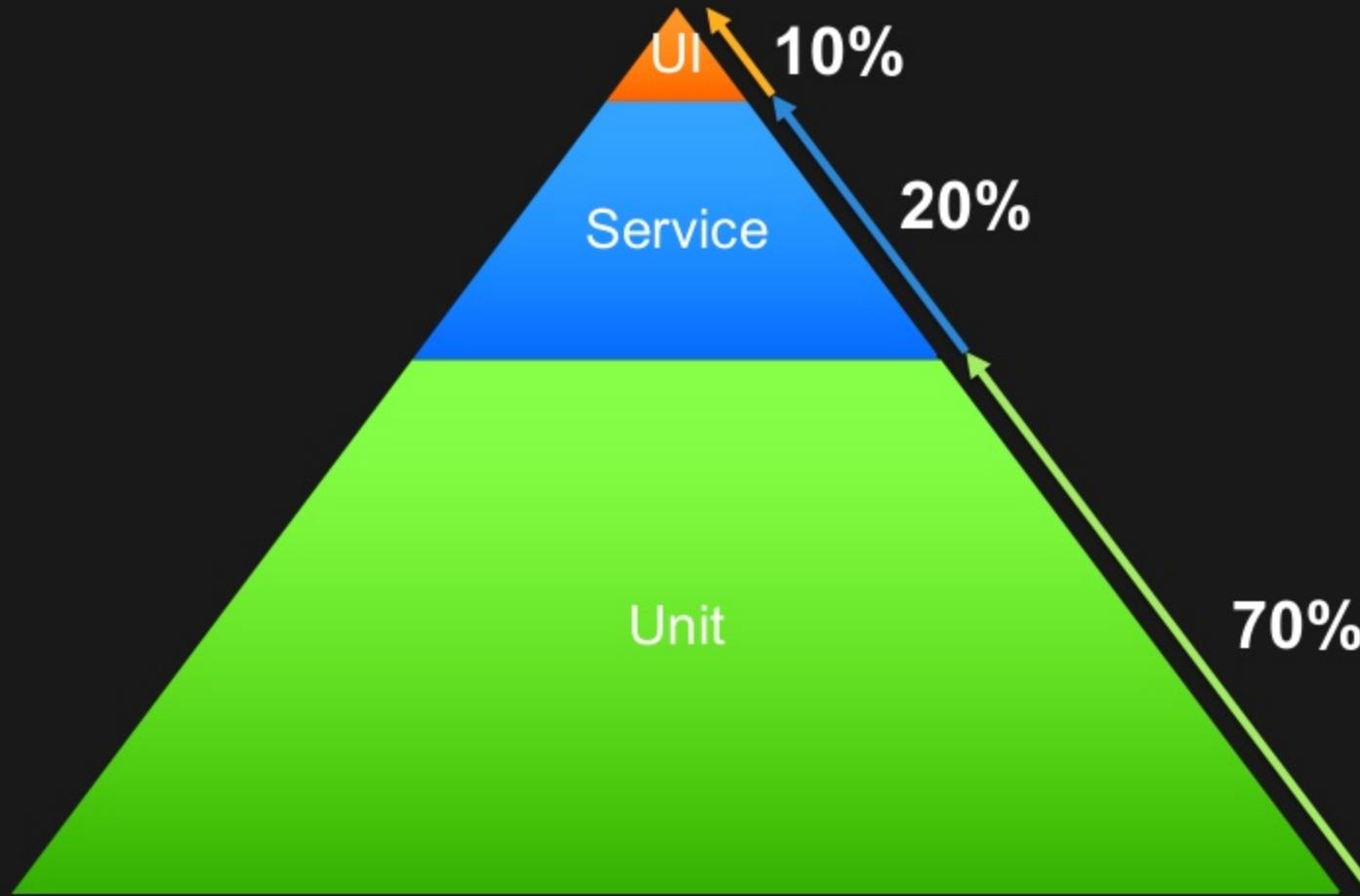
Testing is both a science and an art form!

Goals for testing your code:

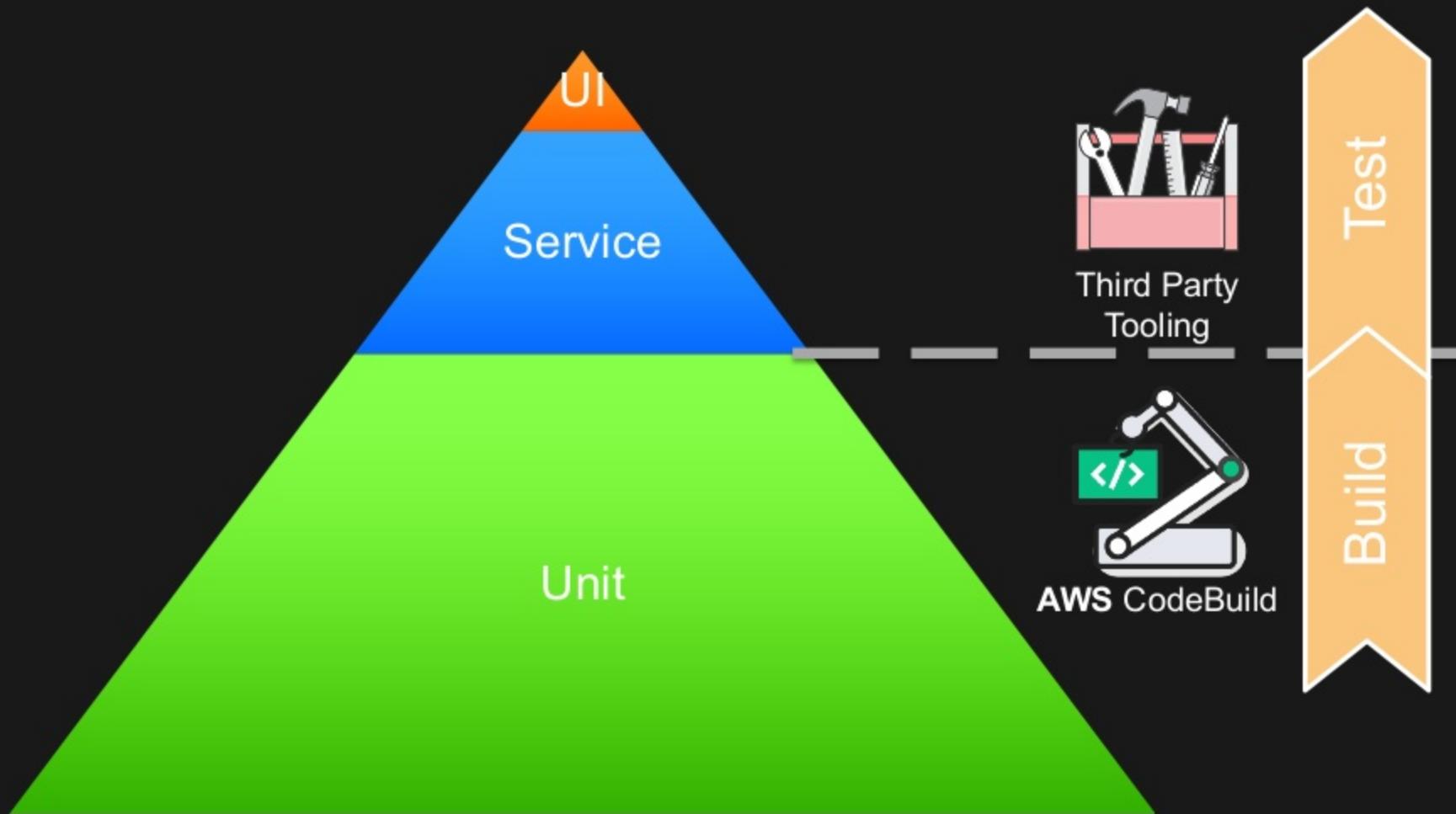
- Want to confirm desired functionality
- Catch programming syntax errors
- Standardize code patterns and format
- Reduce bugs due to non-desired application usage and logic failures
- Make applications more secure



Where to Focus Your Tests:



What service and release step corresponds with which tests?



Pricing

- Pay by the Minute
- Three compute types differentiated by the amount of memory and CPU resources:

| Compute instance type | Memory (GB) | vCPU | Price per build minute (\$) |
|-----------------------|-------------|------|-----------------------------|
| build.general1.small | 3 | 2 | 0.005 |
| build.general1.medium | 7 | 4 | 0.010 |
| build.general1.large | 15 | 8 | 0.020 |

- Free tier of 100 build minutes

*As of January 20 2017

A large wooden trebuchet, a medieval siege engine, is positioned on a stone pier overlooking a body of water under a cloudy sky. The trebuchet's arm is angled upwards, and its wooden frame and metal hardware are clearly visible. In the background, a stone wall and a few people walking on a grassy area are visible.

Deploying your applications

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

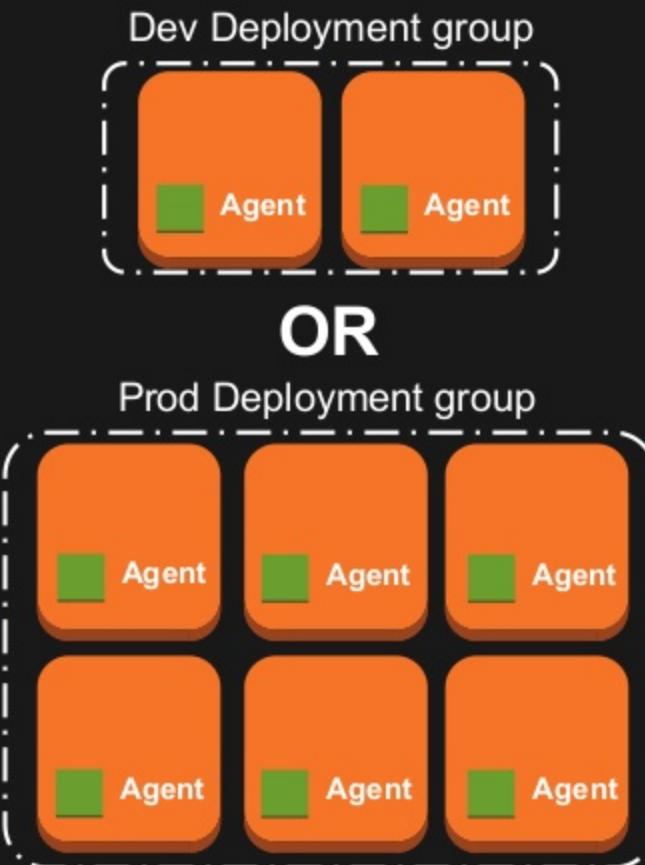
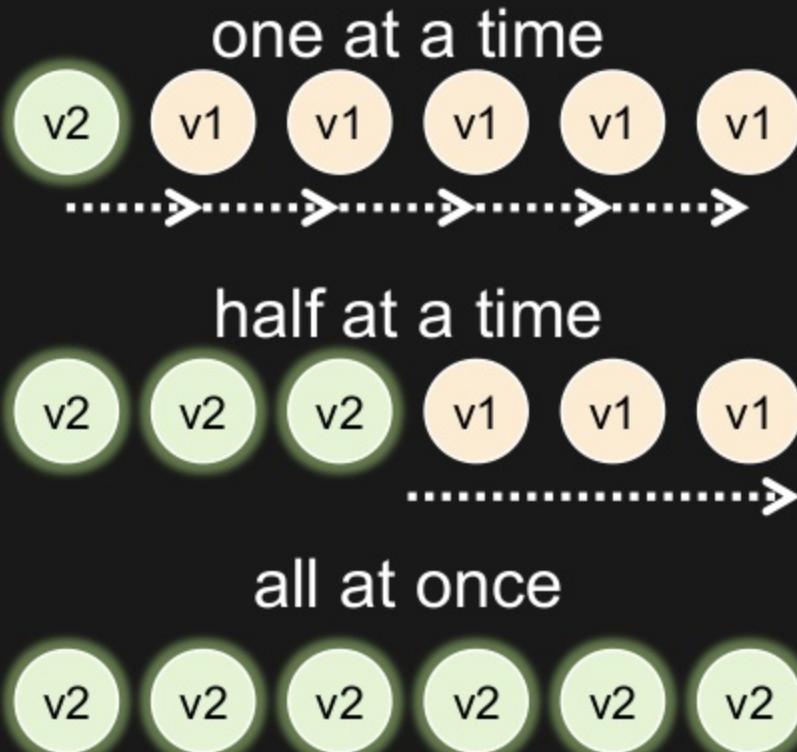
```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
permissions:
  - object: /var/www/html
    pattern: “*.html”
    owner: root
    group: root
    mode: 755
hooks:
  ApplicationStop:
    - location: scripts/deregister_from_elb.sh
  BeforeInstall:
    - location: scripts/install_dependencies.sh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

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!

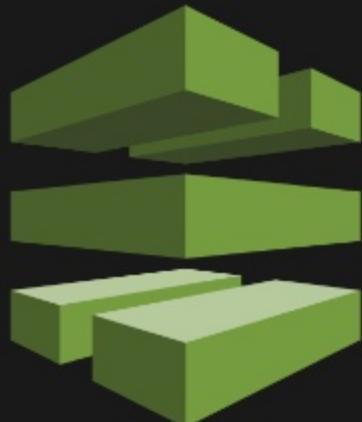
Choose Deployment Speed and Group



Orchestrating build and deploy with a pipeline

AWS CodePipeline

Continuous delivery service for fast and reliable application updates



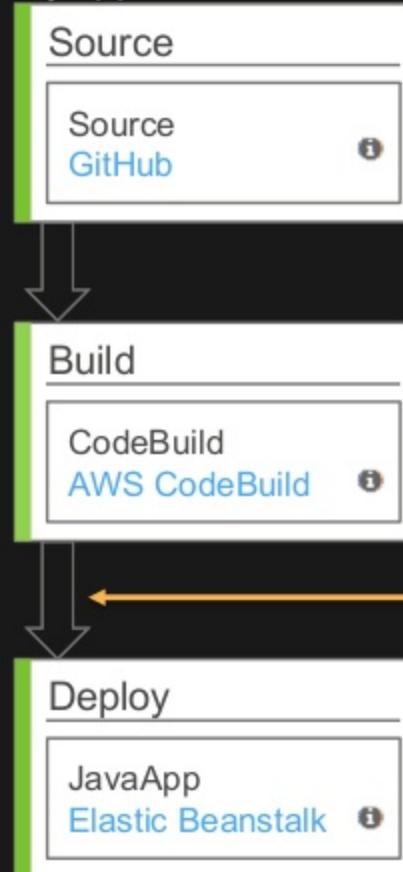
Model and visualize your software release process

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

Integrates with third-party tools and AWS



MyApplication



Stage

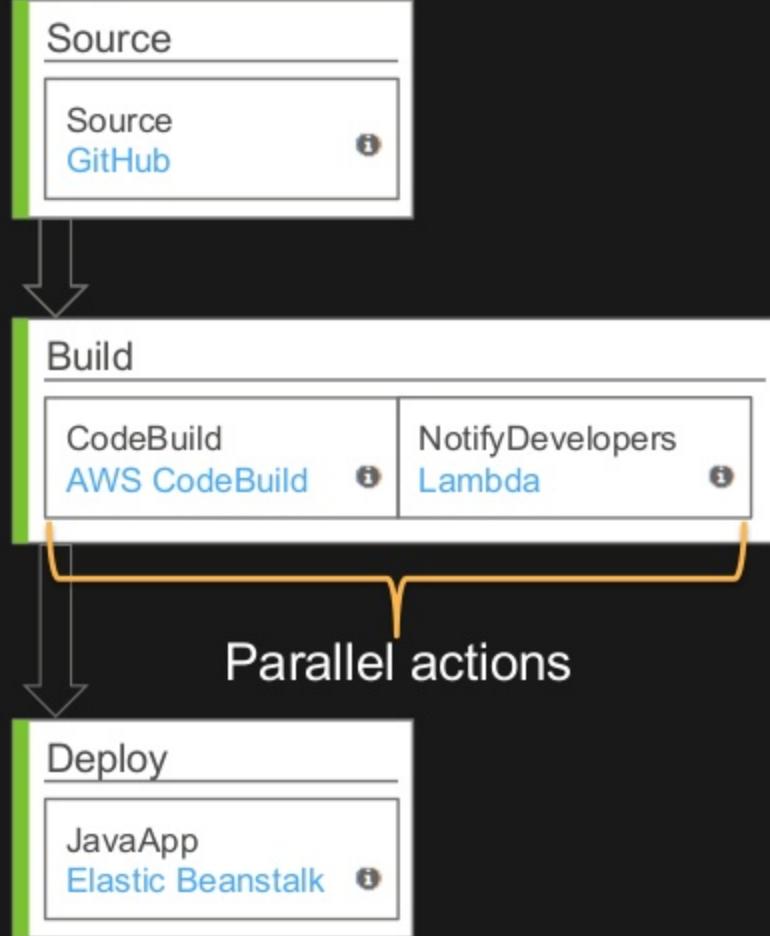
Action

Pipeline

Transition

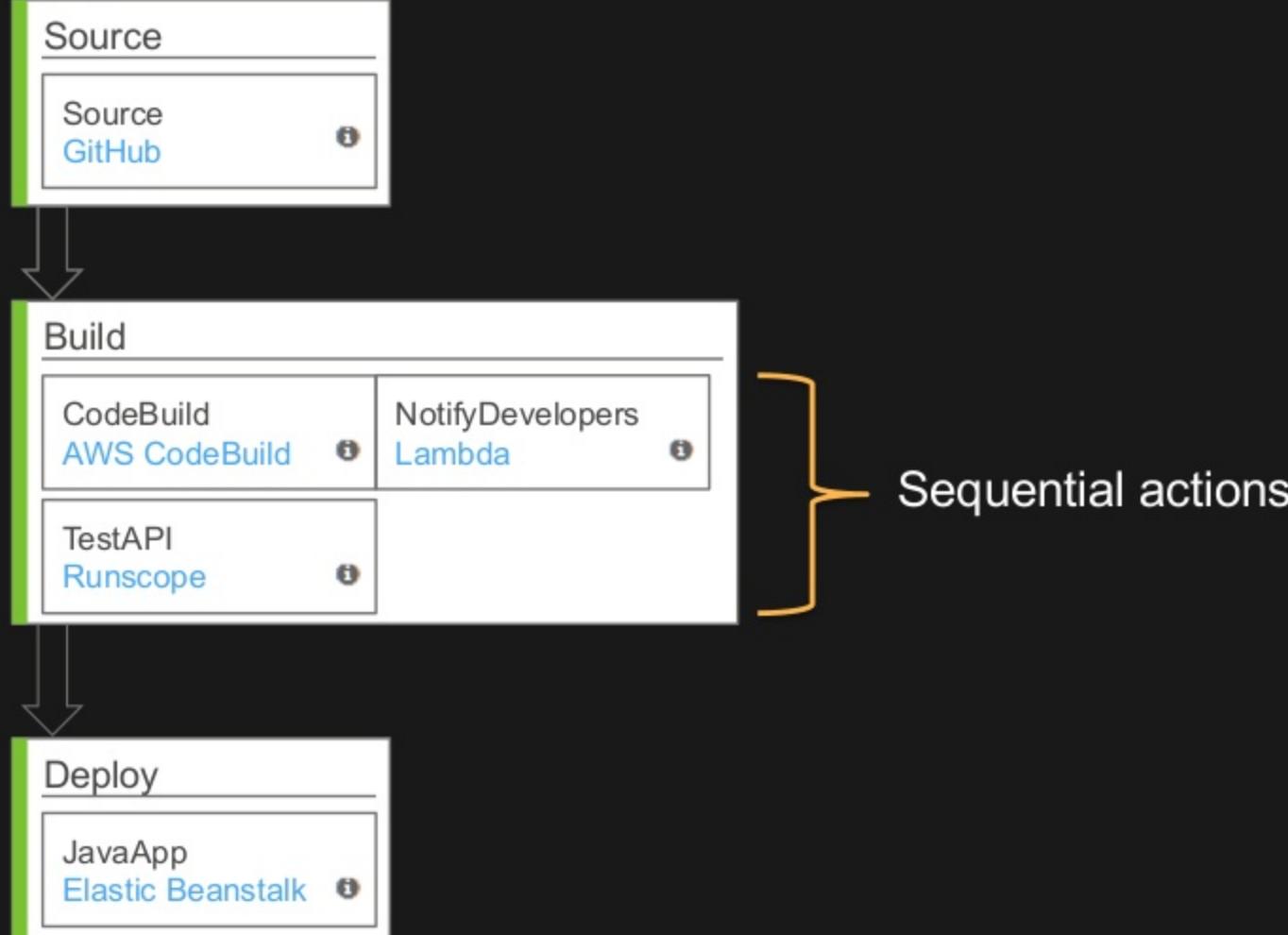


MyApplication



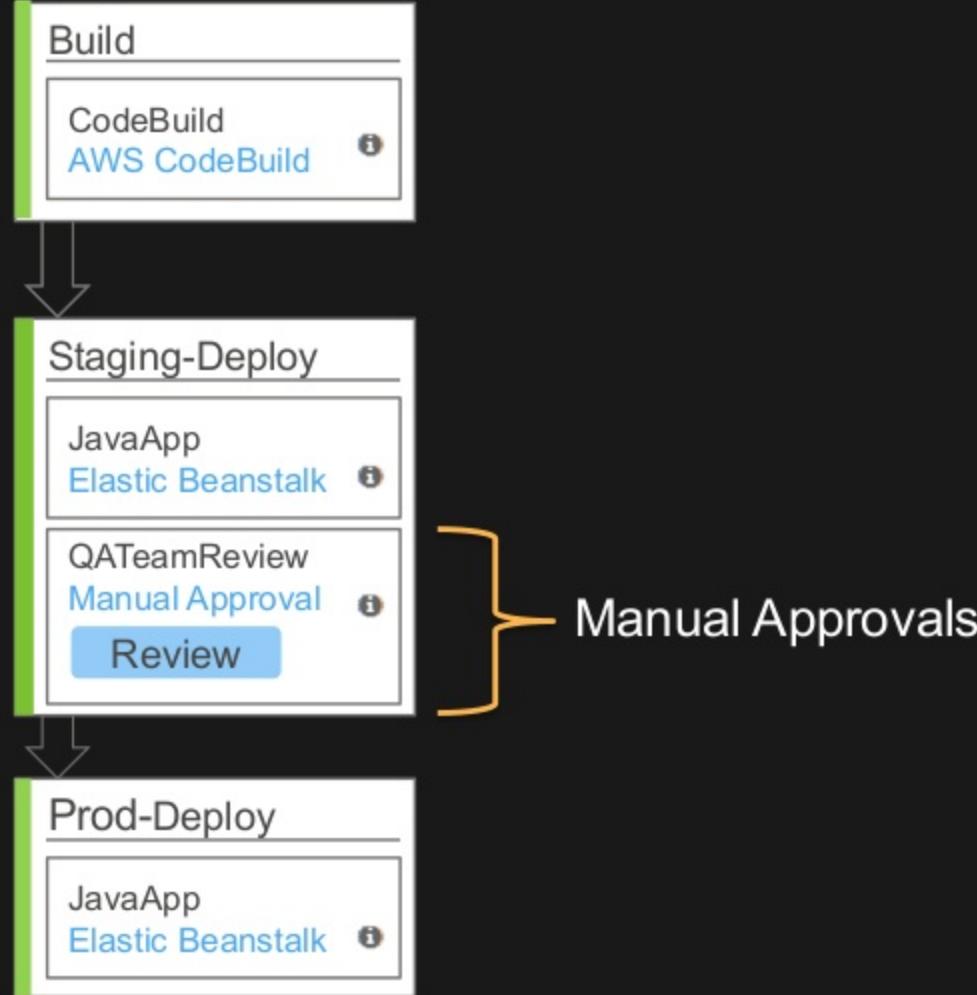


MyApplication





MyApplication



Manual Approvals

AWS CodeCommit



Secure, scalable, and managed Git source control

Use standard Git tools

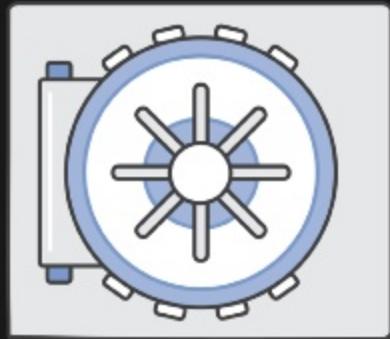
Scalability, availability, and durability of Amazon S3

Encryption at rest with customer-specific keys

No repo size limit

Post commit hooks to call out to SNS/Lambda

Source control in the cloud



Secure



Fully
managed

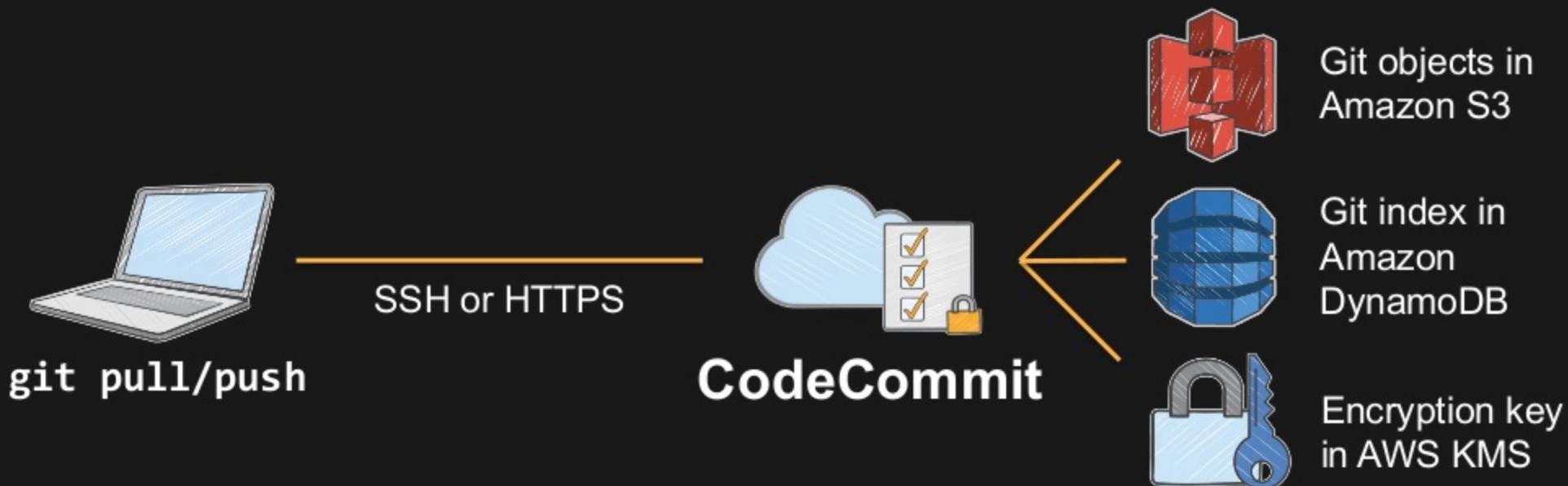


High
availability



Store
anything

AWS CodeCommit



Same Git experience

```
$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/aws-cli
Cloning into 'aws-cli'...
Receiving objects: 100% (16032/16032), 5.55 MiB | 1.25 MiB/s, done.
Resolving deltas: 100% (9900/9900), done.
Checking connectivity... done.
$ nano README.rst
$ git commit -am 'updated README'
[master 4fa0318] updated README
 1 file changed, 1 insertion(+)
$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 297 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote:
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/aws-cli
 4dacd6d..4fa0318  master -> master
```

Pricing

CodeCommit

\$1 per active user per month (first 5 users free)

CodePipeline

\$1 per active pipeline per month (first 1 free)

CodeDeploy

Free to deploy to Amazon EC2

\$0.02 per update to on-prem server

CodeBuild

| Compute Instance Type | Memory(GB) | vCPU | Price per build minute (\$) |
|-----------------------|------------|------|-----------------------------|
| Small | 3 | 2 | 0.005 |
| Medium | 7 | 4 | 0.010 |
| Large | 15 | 8 | 0.020 |

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

AWS CodeStar

Project Templates for EC2, Lambda, and Elastic Beanstalk

AWS CodeStar

Select Template Setup Tools Start Coding

Select an application template to get started. [Help me choose](#)

Refine by

- Application category Web application Web service
- Programming languages Ruby Node.js Java Python PHP HTML 5
- AWS services AWS Elastic Beanstalk Amazon EC2 AWS Lambda

| | | | |
|---|---|---|---|
|  Ruby on Rails Web application <small>AWS Elastic Beanstalk (runs in a managed application environment)</small> |  Ruby on Rails Web application <small>Amazon EC2 (runs on virtual servers that you manage)</small> |  Java Spring Web application <small>AWS Elastic Beanstalk (runs in a managed application environment)</small> |  Java Spring Web application <small>Amazon EC2 (runs on virtual servers that you manage)</small> |
|  Node.js Web application <small>AWS Lambda (running serverless)</small> |  Node.js Web application <small>AWS Elastic Beanstalk (runs in a managed application environment)</small> |  Node.js Web application <small>Amazon EC2 (runs on virtual servers that you manage)</small> |  Python (Django) Web application <small>AWS Elastic Beanstalk (runs in a managed application environment)</small> |
|  Python (Django) Web application |  Express.js Web application |  Express.js Web application |  PHP (Laravel) Web application |

Select Source Control Provider

AWS CodeStar ▶ Create project

Project name

TwoPizzas

Project ID ⓘ

Edit

twopizzas

Which repository do you want to use?

AWS CodeStar will store the project's source code with the service you choose here.



AWS CodeCommit

Highly available Git source control from AWS.
Includes encryption, IAM integration, and more.



GitHub

Creates a GitHub source repository for this project. Requires an existing GitHub account.

Repository name

TwoPizzas

AWS CodeStar

Pre-Configured Continuous Delivery Toolchain

AWS CodeStar

Select Template Setup Tools Start Coding

Project name: AWS SF Summit Project ID: aws-sf-summit Edit

AWS CodeStar includes all of the tools and services you need for a development project.
This project includes:

Source > Build > Test > Deploy > Environment

Release Automation: AWS CodePipeline

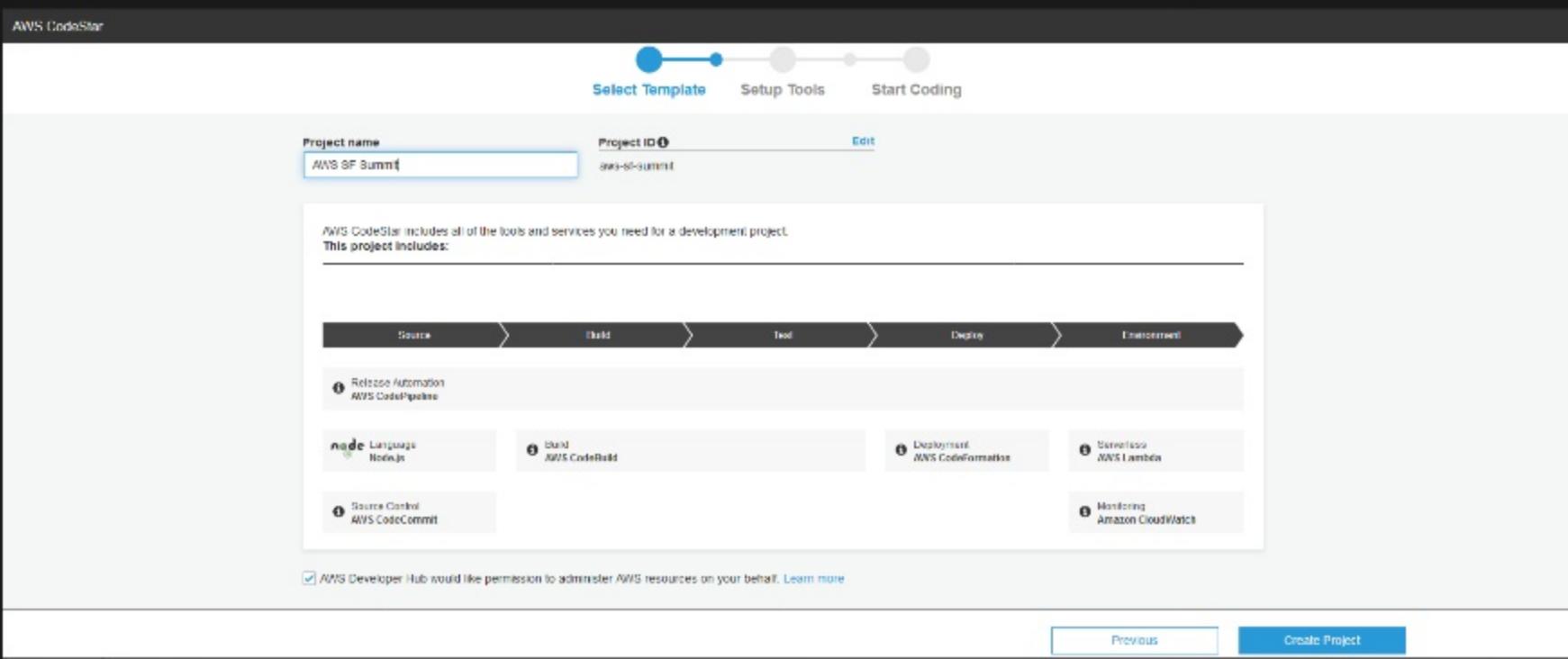
Language: Node.js Build: AWS CodeBuild

Deployment: AWS CodeDeploy Execution: AWS Lambda

Source Control: AWS CodeCommit Monitoring: Amazon CloudWatch

AWS Developer Hub would like permission to administer AWS resources on your behalf. Learn more

Previous Create Project



AWS CodeStar

Easily connect your favorite IDE

AWS CodeStar ▶ Create project



Select template



Set up tools



Start coding

Pick how you want to edit your code



AWS Cloud9

Edit your AWS CodeStar project code with a cloud-based IDE that includes a command line interface. [More info](#)



Command line tools

Edit AWS CodeStar project code by connecting directly to your project's Git source repository.



Eclipse

Configure the AWS Toolkit for Eclipse to edit your AWS CodeStar project code in Eclipse.



Visual Studio

Configure the AWS Toolkit for Visual Studio to edit your CodeStar project code in Microsoft Visual Studio 2015 and later.

AWS CodeStar

Set up secure team access in a few clicks

AWS CodeStar → AWS SF Summit Demo

Project Team

Manage users and permissions in your project.

Add team member

| Display Name | Email Address | Project Role | Remote Access |
|--------------|-------------------------------|--------------|--------------------------|
| user1 | Type email address (Optional) | Viewer | <input type="checkbox"/> |
| user1 | | Contributor | <input type="checkbox"/> |
| user1 | | Owner | <input type="checkbox"/> |

Add

Team member list (3)

| Name | Email | Role | Remote Access | |
|----------------|----------------|-------------|--------------------|---------------|
| C cdhull [You] | cdhull@xyz.com | Owner | Add Public SSH key | Remove Edit |
| A Alex | | Contributor | Not Granted | Remove Edit |
| L luisarias | | Viewer | Not Granted | Remove Edit |

Search team members

AWS CodeStar

Unified Dashboard – Managing Delivery Pipeline and Issue Tracking

The screenshot displays the AWS CodeStar unified dashboard for the project "AWS SF Summit Demo".

Left Sidebar: A vertical sidebar with icons for Dashboard, Code, Build, Deploy, Pipeline, Connect, Team, Extensions, and Project.

Top Navigation: AWS CodeStar > AWS SF Summit Demo. Includes dropdowns for JIRA (My First Project, MFP board, MFP Sprint 1), All Users, and a search bar.

JIRA Issues: A table showing four issues:

| Key | Type | Status | Priority | Issue Summary |
|-------|-------|--------|----------|--|
| MFP-1 | Story | To Do | Medium | Clone the repo |
| MFP-2 | Story | To Do | Medium | Make a change to the sample code |
| MFP-3 | Story | To Do | Medium | Push the change to the project repo |
| MFP-4 | Story | To Do | Medium | Watch my project automatically pick up and deploy the change |

Continuous Deployment Pipeline: Shows three stages: Source (CodeCommit, Succeeded), Build (CodeBuild, Succeeded), and Deploy (CloudFormation, Failed). The Deploy stage failed with the message "ExecuteChangeSet Failed".

Code Samples: Includes "Unit Testing Code Samples" and "User Gamification Feature Testing Samples". You can customize these areas to include links, code samples, or other content useful to your team.

Code Snippet: A snippet of JavaScript code:

```
var increment = function (num) {
  return num++;
};
```

AWS CodeStar

Unified Dashboard – Application Activity and Commit History

Team

Extensions

Project

Application activity

CPUUtilization

18:00 17:00 16:00 15:00 14:00 13:00 12:00 11:00 10:00 09:00 08:00 07:00 06:00 05:00 04:00 03:00 02:00 01:00 00:00 23:00 22:00 21:00 20:00 19:00 18:00 17:00 16:00

0.416
0.17
0.000

CPUUtilization

AWS CloudWatch details

AWS CloudWatch

...

Commit history: sample-project-Repo

Mast

AD Initial commit of sample code made during project creation in AWS DevHub

AWS DevHub committed 1 month ago

780E7b1

Connect

AWS CodeCommit details

Application endpoint(s)

http://dh-w-eben-a4juvlehi36y.eijmply2jy.us-east-1.el...

Continuous deployment

AWS CodePipeline

Source

12/20/2016, 1:53:45 PM

ApplicationSource CodeCommit

Succeeded

Commit history

Application

12/20/2016, 1:55:25 PM

Deploy CodeDeploy

Succeeded

Deploy history

Hosts(1)

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

AWS Cloud9



AWS Cloud9

Cloud-based integrated development environment (IDE)

Let's you write, run, and debug your code with just a browser

Share your environment with your team to pair-program in real-time

Direct terminal access to AWS

Provides great serverless experience: enables local testing and preconfigures the development environment with all SDKs, libraries, and plug-ins

Fully Featured Editor

The screenshot displays the AWS Cloud9 IDE interface with several features highlighted:

- Multiple panels:** The interface shows two code editors side-by-side: `index.js` on the left and `lambda_function.py` on the right.
- Code hinting:** A red callout points to a syntax error in the `lambda_function.py` editor, specifically line 13: `expected an indented block (strring, line 11)`. A tooltip provides the error message.
- Code autocomplete:** A blue callout points to the `index.js` editor, where a tooltip shows the available methods for the `handlers` object, such as `AMAZON.CancelIntent`, `AMAZON.HelpIntent`, etc.

The `index.js` file contains the following code:

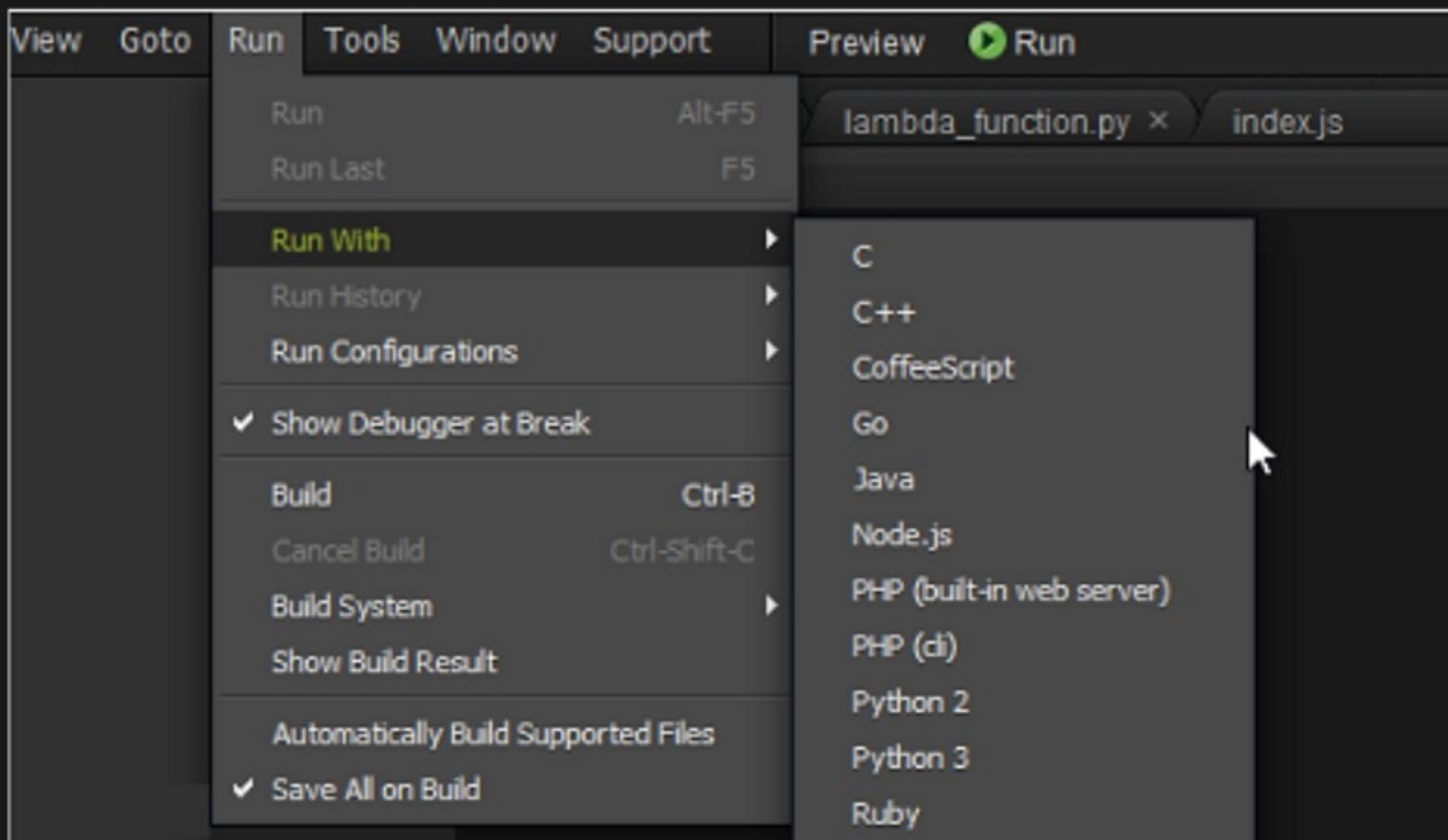
```
88     'Die Temperatur in der Sonne kann 15 Millionen Grad Celsius erreichen.  
89     'Der Mond entfernt sich von unserem Planeten etwa 3,8 cm pro Jahr.',  
90   ],  
91   SKILL_NAME: 'Weltraumwissen auf Deutsch',  
92   GET_FACT_MESSAGE: 'Hier sind deine Fakten: ',  
93   HELP_MESSAGE: 'Du kannst sagen, „Henne mir einen Fakt über den Welt Raum“,  
94   HELP_REPROMPT: 'Wie kann ich dir helfen?',  
95   STOP_MESSAGE: 'Auf Wiedersehen!',  
96 },  
97 );  
98 const handlers = {  
99   'LaunchRequest': function () {  
100     this.emit('GetFact');  
101   },  
102   'GetNewFactIntent': function () {  
103     this.emit('GetFact');  
104   },  
105   'AMAZON.CancelIntent': function () {  
106     this.emit(':ask', 'Okay, see you later!');  
107   },  
108   'AMAZON.HelpIntent': function () {  
109     const speechOutput = this.t('HELP_MESSAGE');  
110     const reprompt = this.t('HELP_MESSAGE');  
111     this.emit(':ask', speechOutput, reprompt);  
112   },  
113   'AMAZON.FallbackIntent': function () {  
114     const speechOutput = this.t('HELP_MESSAGE');  
115     const reprompt = this.t('HELP_MESSAGE');  
116     this.emit(':ask', speechOutput, reprompt);  
117   },  
118 };
```

The `lambda_function.py` file contains the following code:

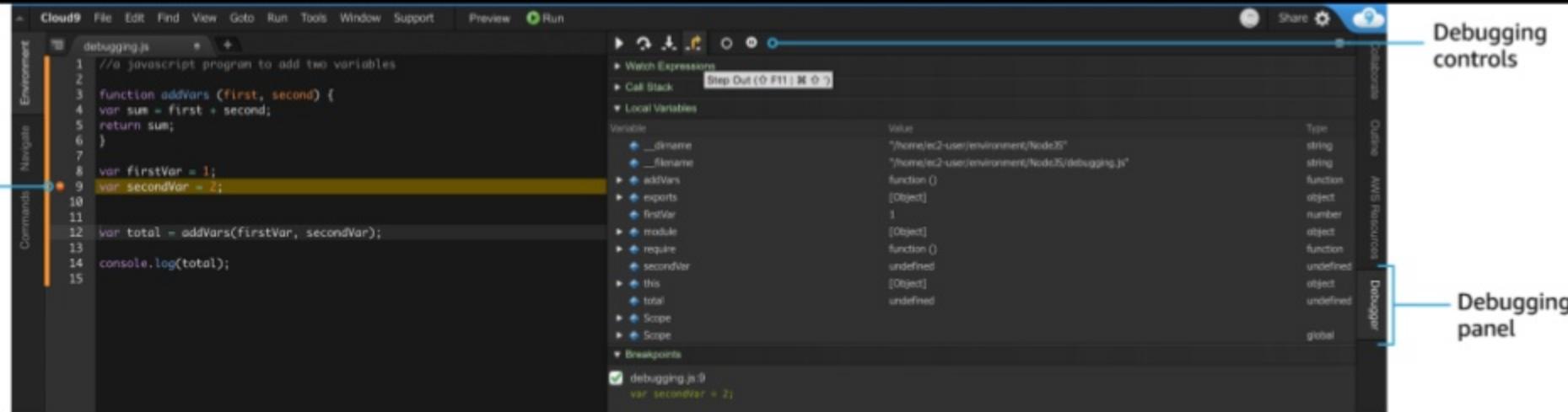
```
1 import json  
2 import urllib  
3 import boto3  
4  
5 print('Loading function')  
6  
7 s3 = boto3.client('s3')  
8  
9 def obcd (dict, dd):  
10  
11 def lambda_handler(event, context):  
12     #print("Received event: " + json.dumps(event, indent=2))  
13     expected an indented block (strring, line 11)  
14     # Get the object from the event and show its content type  
15     bucket = event['Records'][0]['s3']['bucket']['name']  
16     key = urllib.unquote_plus(event['Records'][0]['s3']['object']['key']).decode('utf8')  
17     try:  
18         response = s3.get_object(Bucket=bucket, Key=key)  
19         print("CONTENT TYPE: " + response['ContentType'])  
20         return response['ContentType']  
21     except Exception as e:  
22         print(e)  
23         print('Error getting object {} from bucket {}. Make sure they exist and your bucket has the correct CORS configuration.'.format(key, bucket))  
24         raise e  
25  
26     pri
```

At the bottom, the status bar shows the file types: `99:17 JavaScript Spaces: 4` and `89 Bytes`.

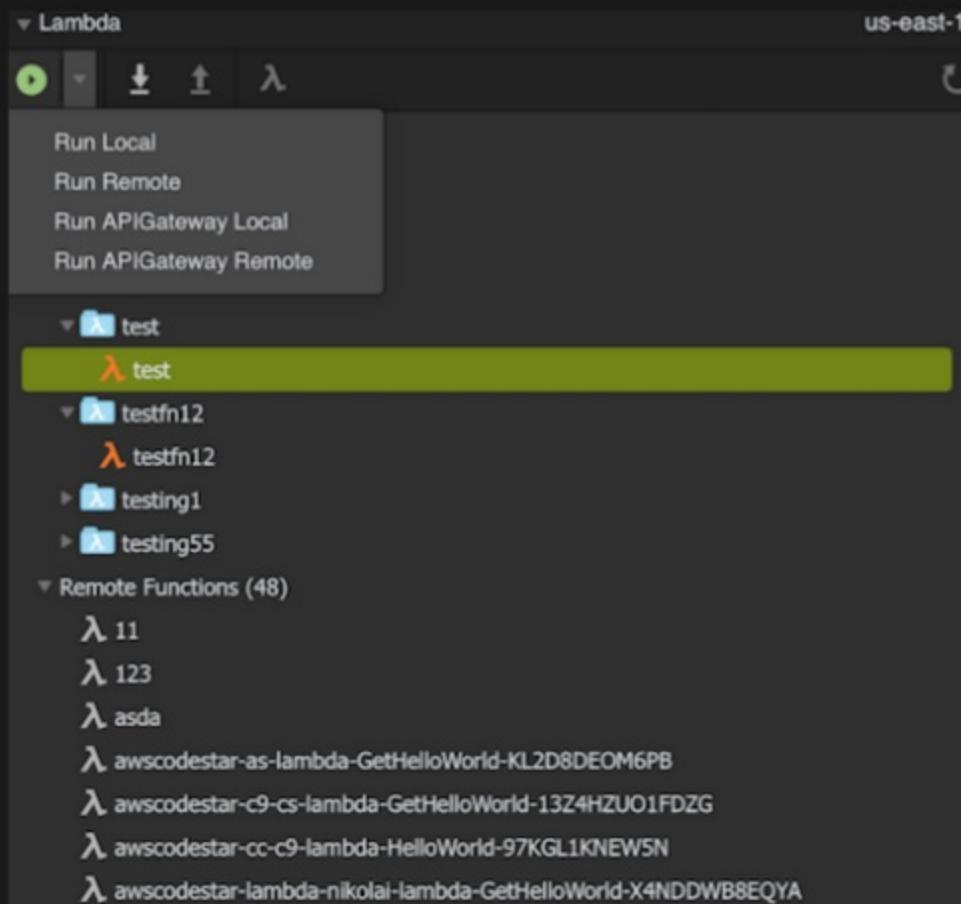
Broad Selection of Run Times



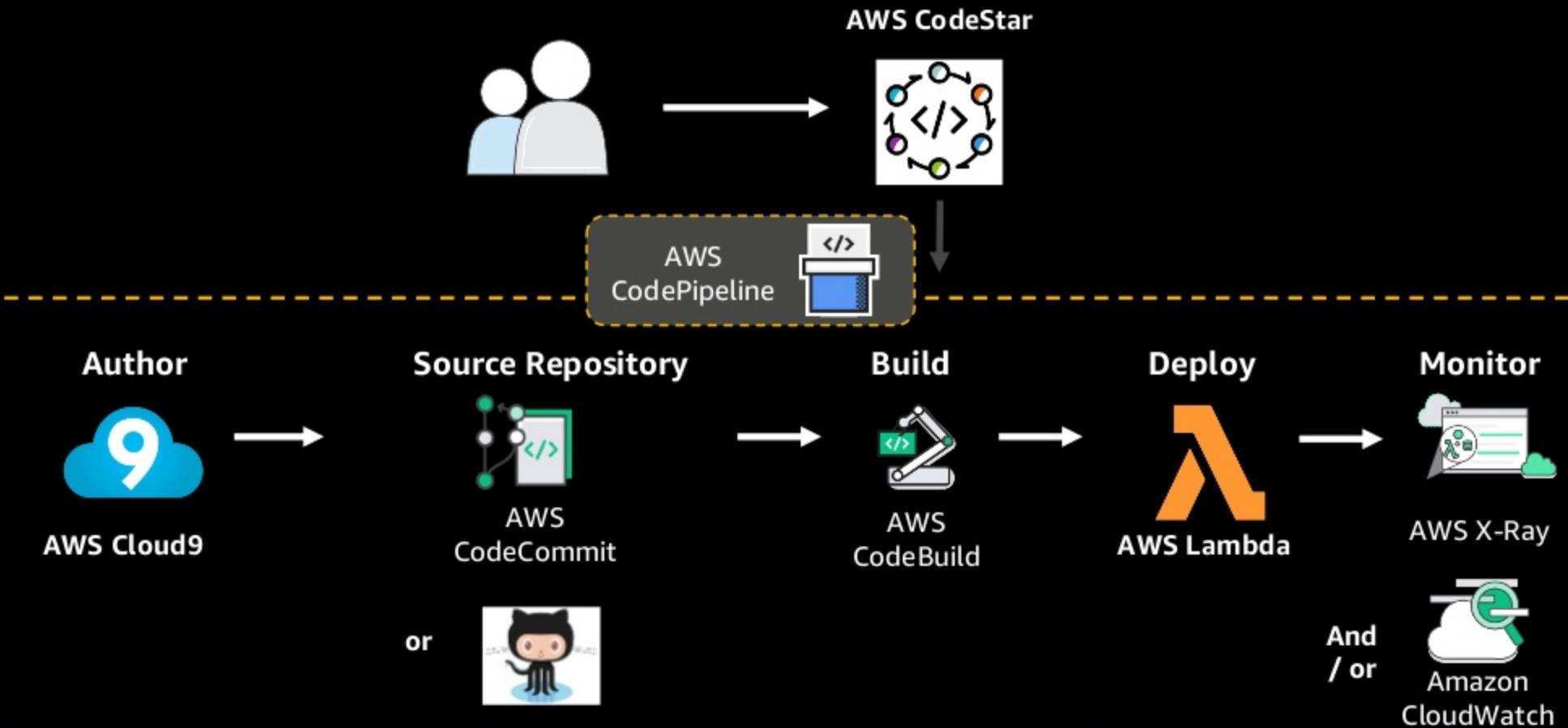
Fully Featured Debugger



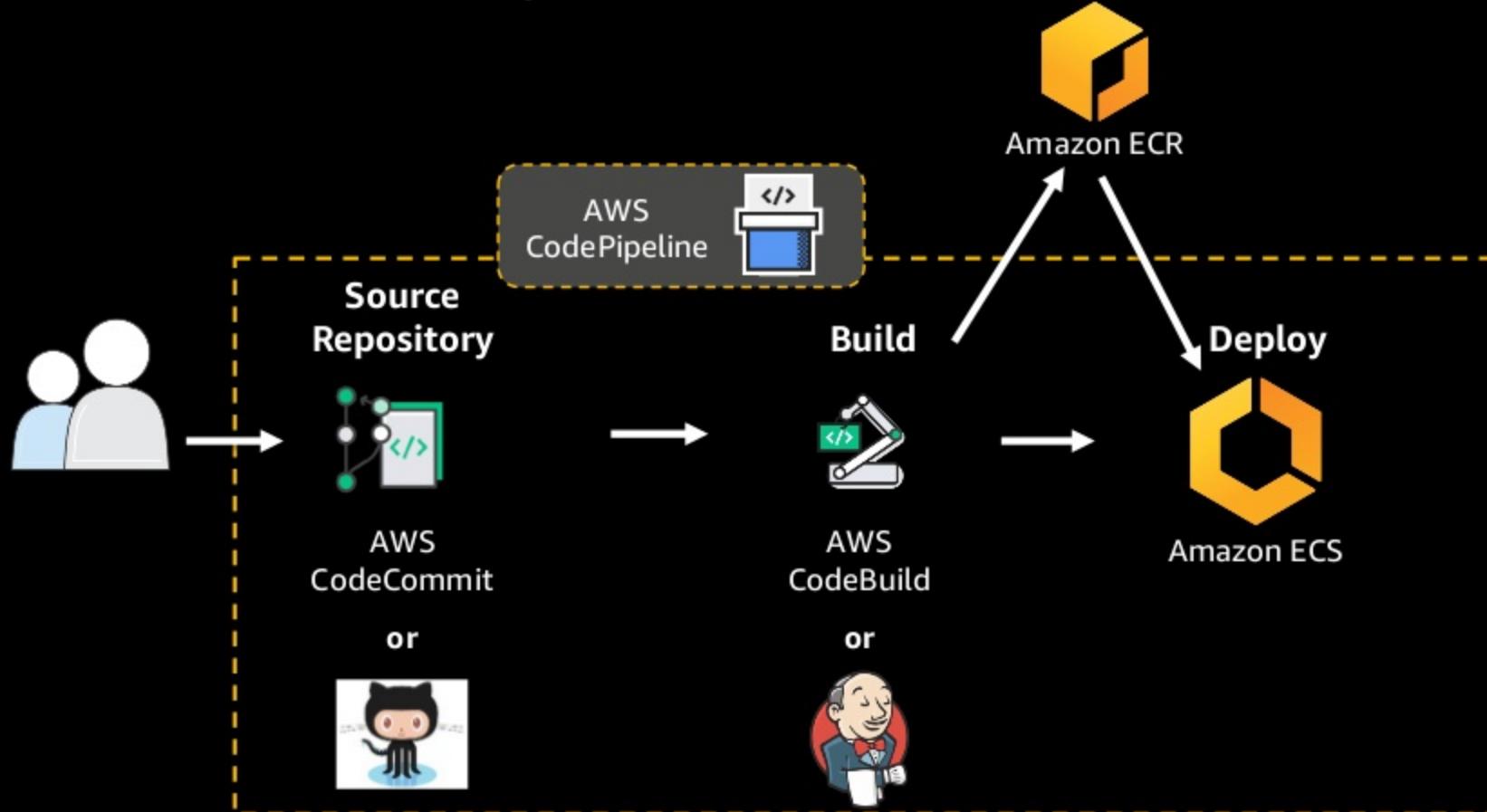
Integrated Tools for Serverless Development



Continuous Delivery for Serverless Applications



Continuous Delivery for Containers

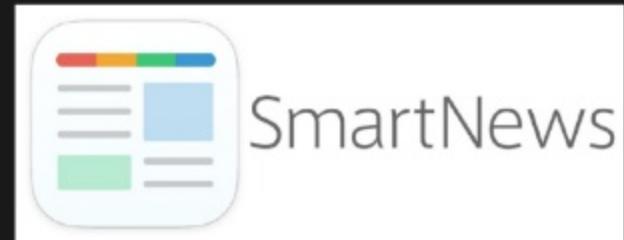


AWS Code featured customers



Instacart

Burt.



GILT



THANK YOU

Appendix

General Best Practices used by Amazon Developers

CI/CD is a MUST!



Everything that is code (application, infrastructure, documentation) goes into a repository

Start with continuous delivery (“gated” promotion) and build up to continuous deployment once evidence of a high-level of excellence in testing is clear

Deploy to canaries, test, deploy to an AZ, test, deploy to a Region, test

General Best Practices used by Amazon Developers (contd.)



Code Reviews are one of the best mechanisms for “good” code:

- Does this code look clean and can someone else understand it?
- Is the design of it meeting the expectations of its needs?

Style checkers

- Will someone else in the company be able to update/fix/maintain this code?

Auto-rollbacks can be the quickest recovery mechanism after failure

- Rollback first, then debug what went wrong with logs/graphs/etc.

AWS Code integrated partners

GitHub

 **Atlassian**

 **Travis CI**


CHEF™

 **Jenkins**

 **CloudBees**

 **circleci**


**puppet
labs**

 **Solano Labs**

 **CODESHIP**

 **XebiaLabs**
Deliver Faster


ANSIBLE

 **Apica**

 **Runscope**

 **Ghost Inspector**

 **BlazeMeter**

 **TeamCity**



 **HASHICORP™**

 **SALTSTACK**

AWS DevOps consulting partners



CLOUDREACH



logicworks



REĀN
CLOUD

slalom



stelligent

TriNimbus
Cloud Management Solutions

DEMO!

Demo:

1. Start with a repository (github.com/awslabs/aws-codedeploy-sample-tomcat)
2. Add buildspec.yml
3. Create CodePipeline pipeline with a Source and Build stage
4. Do a build
5. Add a deploy stage
6. Do a full execution of the pipeline

[Code](#)[Issues 1](#)[Pull requests 2](#)[Projects 0](#)[Pulse](#)[Graphs](#)[Settings](#)

A sample Tomcat application integrated with CodeDeploy.

[Edit](#)[15 commits](#)[1 branch](#)[0 releases](#)[6 contributors](#)[Apache-2.0](#)Branch: [master](#) ▾[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#) ▾

| | | |
|--|--|--------------------------------------|
|  teknogeek0 committed on GitHub | Create buildspec.yml | Latest commit bba931d 11 minutes ago |
|  scripts | Bumping Tomcat to 7.0.73 | 21 minutes ago |
|  src/main | Import sample application to GitHub. | 11 months ago |
|  .gitignore | Import sample application to GitHub. | 11 months ago |
|  LICENSE | Initial commit | a year ago |
|  README.md | Initial commit | a year ago |
|  appspec.yml | Import sample application to GitHub. | 11 months ago |
|  buildspec.yml | Create buildspec.yml | 11 minutes ago |
|  pom.xml | Declared maven-compiler-version, which was breaking build on 3.3.9 ve... | 9 months ago |

[Code](#)[Issues 1](#)[Pull requests 2](#)[Projects 0](#)[Pulse](#)[Graphs](#)[Settings](#)

A sample Tomcat application integrated with CodeDeploy.

[Edit](#)[15 commits](#)[1 branch](#)[0 releases](#)[6 contributors](#)[Apache-2.0](#)Branch: [master](#) ▾[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#) ▾

| | | |
|--|--|--------------------------------------|
|  teknogeek0 committed on GitHub | Create buildspec.yml | Latest commit bba931d 11 minutes ago |
|  scripts | Bumping Tomcat to 7.0.73 | 21 minutes ago |
|  src/main | Import sample application to GitHub. | 11 months ago |
|  .gitignore | Import sample application to GitHub. | 11 months ago |
|  LICENSE | Initial commit | a year ago |
|  README.md | Initial commit | a year ago |
|  appspec.yml | Import sample application to GitHub. | 11 months ago |
|  buildspec.yml | Create buildspec.yml | 11 minutes ago |
|  pom.xml | Declared maven-compiler-version, which was breaking build on 3.3.9 ve... | 9 months ago |

[Code](#)[Issues 1](#)[Pull requests 2](#)[Projects 0](#)[Pulse](#)[Graphs](#)[Settings](#)Branch: [master](#) ▾[aws-codedeploy-sample-tomcat / buildspec.yml](#)[Find file](#)[Copy path](#) **teknogeek0** Create buildspec.yml

bba931d 12 minutes ago

1 contributor

12 lines (9 sloc) | 138 Bytes

[Raw](#)[Blame](#)[History](#)

```
1 version: 0.1
2
3 phases:
4   build:
5     commands:
6       - echo Build completed on `date`
7       - mvn package
8
9 artifacts:
10  files:
11    - '**/*'
```

Create pipeline

Step 1: Name[Step 2: Source](#)[Step 3: Build](#)[Step 4: Beta](#)[Step 5: Service Role](#)[Step 6: Review](#)

Getting started with AWS CodePipeline



These steps will help you set up your first pipeline. Begin by giving your pipeline a name.

Pipeline name*

JustCI

[Cancel](#)[Next step](#)*** Required**[Step 1: Name](#)**Step 2: Source**[Step 3: Build](#)[Step 4: Beta](#)[Step 5: Service Role](#)[Step 6: Review](#)

Source location



Specify where your source code is stored. Choose the provider, and then provide connection details for that provider.

Source provider*

GitHub

Connect to GitHub

Choose a repository from the list of repositories, and then select the branch you want to use. You must be an administrator of the repository. [Learn more](#)

Repository*

tekno geek0/ aws-codedeploy-sample-tomcat

Branch*

master

*** Required**[Cancel](#)[Previous](#)[Next step](#)



Create pipeline

[Step 1: Name](#)[Step 2: Source](#)**Step 3: Build**[Step 4: Beta](#)[Step 5: Service Role](#)[Step 6: Review](#)

Build



Choose the build provider that you want to use or that you are already using.

Build provider*

AWS CodeBuild

AWS CodeBuild

AWS CodeBuild is a fully managed build service that builds and tests code in the cloud. CodeBuild scales continuously. You only pay by the minute. [Learn more](#)

Configure your project

- Select an existing build project
- Create a new build project

Project name*

CiDemo

**Description** [+ Add description](#)

AWS CodePipeline Management Console

Create pipeline

[Step 1: Name](#)[Step 2: Source](#)[Step 3: Build](#)**Step 4: Beta**[Step 5: Service Role](#)[Step 6: Review](#)

Beta



Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

Deployment provider*

No deployment

You are creating the pipeline without a deployment stage. You can edit your pipeline later to add one or more deployment stages.

* Required

[Cancel](#)[Previous](#)[Next step](#)

Create pipeline

[Step 1: Name](#)[Step 2: Source](#)[Step 3: Build](#)[Step 4: Beta](#)**Step 5: Service Role**[Step 6: Review](#)

AWS Service Role



Create a service role in IAM to give AWS CodePipeline permission to use resources in your account. If you already have a service role configured for this purpose, you can choose it from the list instead of creating a role. However, if that role is not configured correctly, AWS CodePipeline might not work as expected.

Role name*

[Create role](#)

* Required

[Cancel](#)[Previous](#)[Next step](#)



Create pipeline

[Step 1: Name](#)[Step 2: Source](#)[Step 3: Build](#)[Step 4: Beta](#)[Step 5: Service Role](#)**Step 6: Review**

Review your pipeline



We will create your pipeline with the following resources.

Source Stage

Source provider GitHub

Repository teknogeek0/aws-codedeploy-sample-tomcat

Branch master

Build Stage

Build provider AWS CodeBuild

Project name* CiDemo [View project details](#)

Beta Stage

Deployment provider No Deployment

JustCI

View progress and manage your pipeline.

[Edit](#)[Release change](#)

Source

Source



GitHub

✓ Succeeded just now f066258 Source: Merge pull requ...

Build

CodeBuild



AWS CodeBuild

⌚ In Progress just now Source: Merge pull requ...

JustCI

View progress and manage your pipeline.

[Edit](#)[Release change](#)

Source

Source



GitHub

✓ Succeeded 3 min ago
f066258 Source: Merge pull requ...

Build

CodeBuild



AWS CodeBuild

✓ Succeeded just now [Details](#) Source: Merge pull requ...



AWS CodeBuild

Build projects

Build history

CiDemo:0535e234-646f-400a-ae3b-d103209bbc4b Succeeded



Review your build details as it progresses.

Build

Build ARN arn:aws:codebuild:us-east-1:102901597367:build/CiDemo:0535e234-646f-400a-ae3b-d103209bbc4b

Build project CiDemo

Source provider AWS CodePipeline

Repository

Start time 4 minutes ago

End time 2 minutes ago

Status Succeeded

Initiator JustCI

▶ Build details

Phase details

| | Name | Status | Duration | Completed |
|---|-----------------|--|----------|---------------|
| ▶ | SUBMITTED | Succeeded | - | 4 minutes ago |
| ▶ | PROVISIONING | Succeeded | 59 secs | 3 minutes ago |
| ▶ | DOWNLOAD_SOURCE | Succeeded | 13 secs | 2 minutes ago |
| ▶ | PRE_BUILD | Succeeded | - | 2 minutes ago |

 Expand all Row Text

0

6

1

Filter events

all 30s 5m 1h 6h 1d 1w cu

| | Time (UTC +00:00) | Message |
|---|-------------------|--|
| | 2017-01-20 | |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/SampleMavenTomcatApp/WEB-INF/web.xml |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/SampleMavenTomcatApp.war |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/classes/com/amazonaws/labs/sampleapp/IndexController.class |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/classes/com/amazonaws/labs/sampleapp/MvcConfiguration.class |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/classes/css/theme.css |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/classes/images/alterations-dog4.png |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/classes/images/casual-dog2.png |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/classes/images/classy-dog1.png |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/classes/images/fancy-dog3.png |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/classes/images/sales-lobsterdog.png |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/maven-archiver/pom.properties |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/maven-status/maven-compiler-plugin/compile/default-compile/createdFiles |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Found target/maven-status/maven-compiler-plugin/compile/default-compile/inputFiles |
| ▶ | 05:33:15 | [Container] 2017/01/20 05:33:14 Creating zip artifact |
| ▶ | 05:33:19 | [Container] 2017/01/20 05:33:15 Phase complete: UPLOAD_ARTIFACTS Success: true |
| ▶ | 05:33:19 | [Container] 2017/01/20 05:33:15 Phase context status code: Message: |

Demo:

- ✓ Start with a repository (github.com/awslabs/aws-codedeploy-sample-tomcat)
- ✓ Add buildspec.yml
- ✓ Create CodePipeline pipeline with a Source and Build stage
- ✓ Do a build
 - Add a deploy stage
 - Do a full execution of the pipeline

[Code](#)[Issues 1](#)[Pull requests 2](#)[Projects 0](#)[Pulse](#)[Graphs](#)[Settings](#)

A sample Tomcat application integrated with CodeDeploy.

[Edit](#)[15 commits](#)[1 branch](#)[0 releases](#)[6 contributors](#)[Apache-2.0](#)Branch: [master](#) ▾[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download](#) ▾

| | | |
|--|--|--------------------------------------|
|  teknogeek0 committed on GitHub | Create buildspec.yml | Latest commit bba931d 11 minutes ago |
|  scripts | Bumping Tomcat to 7.0.73 | 21 minutes ago |
|  src/main | Import sample application to GitHub. | 11 months ago |
|  .gitignore | Import sample application to GitHub. | 11 months ago |
|  LICENSE | Initial commit | a year ago |
|  README.md | Initial commit | a year ago |
|  appspec.yml | Import sample application to GitHub. | 11 months ago |
|  buildspec.yml | Create buildspec.yml | 11 minutes ago |
|  pom.xml | Declared maven-compiler-version, which was breaking build on 3.3.9 ve... | 9 months ago |

Branch: master ▾

[aws-codedeploy-sample-tomcat / appspec.yml](#)[Find file](#)[Copy path](#)

Jmcfar Import sample application to GitHub.

2a4178b on Mar 7, 2016

1 contributor

21 lines (20 sloc) 600 Bytes

[Raw](#)[Blame](#)[History](#)

```
1 version: 0.0
2 os: linux
3 files:
4   - source: /target/SampleMavenTomcatApp.war
5     destination: /tmp/codedeploy-deployment-staging-area/
6   - source: /scripts/configure_http_port.xsl
7     destination: /tmp/codedeploy-deployment-staging-area/
8 hooks:
9   ApplicationStop:
10    - location: scripts/stop_application
11      timeout: 300
12   BeforeInstall:
13    - location: scripts/install_dependencies
14      timeout: 300
15   ApplicationStart:
16    - location: scripts/write_codedeploy_config.sh
17    - location: scripts/start_application
18      timeout: 300
19   ValidateService:
20    - location: scripts/basic_health_check.sh
```

aws-codedeploy-sample-tomcat/appspec.yml at master · awslabs/aws-codedeploy-sample-to



AWS CodePipeline

Edit: JustCI

Add or edit a stage in a pipeline or actions in a stage. [Learn more](#)

[Cancel](#)[Delete](#)[Save pipeline changes](#)

Source

Source
GitHub



+ Stage

Build

CodeBuild
AWS CodeBuild



+ Stage

Staging



Add action

Deployment provider*

AWS CodeDeploy

AWS CodeDeploy

Choose one of your existing applications, or [create a new one in AWS CodeDeploy](#).

Application name*

jan20screen



Choose one of your existing deployment groups, or [create a new one in AWS CodeDeploy](#).

Deployment group*

jan20screen-DevEnv



Input artifacts

Choose one or more input artifacts for this action. The output of previous actions can be the input of this action. [Learn more](#)

* Required

[Cancel](#)[Add action](#)

Deployment group*

jan20screen-DevEnv



Input artifacts

Source

Source
GitHub



Stage

Build

CodeBuild
AWS CodeBuild



Stage

Staging

Action

StagingDeploy
AWS CodeDeploy



Action

Action

#

JustCI

View progress and manage your pipeline.

[Edit](#)[Release change](#)

Source

Source

GitHub

✓ Succeeded 8 min ago
f066258

Source: Merge pull requ...

Build

CodeBuild

AWS CodeBuild

✓ Succeeded 5 min ago [Details](#)

Source: Merge pull requ...

Staging

StagingDeploy

AWS CodeDeploy

⌚ In Progress 1 min ago [Details](#)

Source: Merge pull requ...

JustCI

View progress and manage your pipeline.

[Edit](#)[Release change](#)

Source

Source

GitHub

✓ Succeeded 8 min ago
f066258

Source: Merge pull requ...

Build

CodeBuild

AWS CodeBuild

✓ Succeeded 5 min ago [Details](#)

Source: Merge pull requ...

Staging

StagingDeploy

AWS CodeDeploy

✓ Succeeded just now [Details](#)

Source: Merge pull requ...

Demo:

- ✓ Start with a repository (github.com/aws-labs/aws-codedeploy-sample-tomcat)
- ✓ Add buildspec.yml
- ✓ Create CodePipeline pipeline with a Source and Build stage
- ✓ Do a build
- ✓ Add a deploy stage
- ✓ Do a full execution of the pipeline

General Best Practices used by Amazon Developers

- CI/CD is a MUST!
 - Commit frequently
 - Builds on every commit
 - Build once in a given execution flow
 - Deploy to a running environment for further testing



General Best Practices used by Amazon Developers

- CI/CD is a MUST!
 - Commit frequently
 - Builds on every commit
 - Build once in a given execution flow
 - Deploy to a running environment for further testing
- Everything that is code (application, infrastructure, documentation) goes into a repository
 - If its not in a repository, it doesn't go into Production environments!



General Best Practices used by Amazon Developers

- CI/CD is a MUST!
 - Commit frequently
 - Builds on every commit
 - Build once in a given execution flow
 - Deploy to a running environment for further testing
- Everything that is code (application, infrastructure, documentation) goes into a repository
 - If its not in a repository, it doesn't go into production environments!
- Start with continuous delivery (“gated” promotion) and build up to continuous deployment once evidence of a high-level of excellence in testing is clear



General Best Practices used by Amazon Developers

- CI/CD is a MUST!
 - Commit frequently
 - Builds on every commit
 - Build once in a given execution flow
 - Deploy to a running environment for further testing
- Everything that is code (application, infrastructure, documentation) goes into a repository
 - If its not in a repository, it doesn't go into production environments!
- Start with continuous delivery (“gated” promotion) and build up to continuous deployment once evidence of a high-level of excellence in testing is clear
- Deploy to canaries, test, deploy to an AZ, test, deploy to a Region, test



General Best Practices used by Amazon Developers (cont.)

- Code Reviews are one of the best mechanisms for “good” code:
 - Does this code look clean and can someone else understand it?
 - Is the design of it meeting the expectations of its needs?
 - Are there better/easier ways to do this same thing?



General Best Practices used by Amazon Developers (cont.)

- Code Reviews are one of the best mechanisms for “good” code:
 - Does this code look clean and can someone else understand it?
 - Is the design of it meeting the expectations of its needs?
 - Are there better/easier ways to do this same thing?
- Style checkers
 - Will someone else in the company be able to update/fix/maintain this code?



General Best Practices used by Amazon Developers (cont.)

- Code Reviews are one of the best mechanisms for “good” code:
 - Does this code look clean and can someone else understand it?
 - Is the design of it meeting the expectations of its needs?
 - Are there better/easier ways to do this same thing?
- Style checkers
 - Will someone else in the company be able to update/fix/maintain this code?
- Auto-rollbacks can be the quickest recovery mechanism after failure
 - Rollback first, then debug what went wrong with logs/graphs/etc.



General Best Practices used by Amazon Developers (cont.)

- Code Reviews are one of the best mechanisms for “good” code:
 - Does this code look clean and can someone else understand it?
 - Is the design of it meeting the expectations of its needs?
 - Are there better/easier ways to do this same thing?
- Style checkers
 - Will someone else in the company be able to update/fix/maintain this code?
- Auto-rollbacks can be the quickest recovery mechanism after failure
 - Rollback first, then debug what went wrong with logs/graphs/etc.
- Thorough dashboards
 - What is happening now?
 - What “normal” looks like typically over some period of time?
 - What do I do if this graph looks wrong/an alarm has been triggered?
 - What events can I correlate with a move in a graph?



Code* Tips and Tricks

- All Code* products can(and should) be provisioned and managed with AWS CloudFormation!
 - You could literally store the CloudFormation templates that provision your Code* resources in CodeCommit and update them via CodePipeline (It's like Code* Inception!)
- Deep integration with IAM. You can assign permissions on who can commit code, approve manual approvals, deploy to certain deployment groups and more!
- Integrate with AWS Lambda to do almost anything:
 - CodeCommit has Repository Triggers
 - CodeDeploy has Event Notifications
 - CodePipeline has native Lambda invoke





DevOps and AWS

Tooling and infrastructure resources for DevOps practitioners

[Get Started with AWS](#)[What is DevOps?](#)[DevOps Blog](#)[Partner Solutions](#)[Resources](#)

AWS provides a set of flexible services designed to enable companies to more rapidly and reliably build and deliver products using AWS and DevOps practices. These services simplify provisioning and managing infrastructure, deploying application code, automating software release processes, and monitoring your application and infrastructure performance.

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management

AWS DevOps Blog

≡ Menu



AWS re:Invent

Products

More ▾

My Account ▾

Sign In to the Console

AWS DevOps Blog

Building a Cross-Region/Cross-Account Code Deployment Solution on AWS

by BK Chaurasiya | on 01 NOV 2016 | in [How-To](#) | [Permalink](#) | [Comments](#)



Many of our customers have expressed a desire to build an end-to-end release automation workflow solution that can deploy changes across multiple regions or different AWS accounts.

In this post, I will show you how you can easily build an automated cross-region code deployment solution using [AWS CodePipeline](#) (a continuous delivery service), [AWS CodeDeploy](#) (an automated application deployment service), and [AWS Lambda](#) (a serverless compute service). In the Taking This Further section, I will also show you how to extend what you've learned so that you can create a cross-account deployment solution.

We will use AWS CodeDeploy and AWS CodePipeline to create a multi-pipeline solution running in two regions (Region A and Region B). Any update to the source code in Region A will trigger validation and deployment of source code changes in the pipeline in Region A. A successful processing of source code in all of its AWS CodePipeline stages will invoke a Lambda function as a custom action, which will copy the source code into an S3 bucket in Region B. After the source code is copied into this bucket, it will trigger a similar chain of processes into the different AWS CodePipeline stages in Region B. See the following diagram.

Search the DevOps Blog

Search

Resources

[AWS Developer Tools](#)

[AWS Management Tools](#)

[DevOps and AWS](#)

[Resources for DevOps on AWS](#)

AO..

QUESTION
EVERYTHING

"TEASEF

TLW