

Practicing CI/CD with AWS CodePipeline

BUILDING AND DEPLOYING WITH CODEPIPELINE



Ryan Lewis

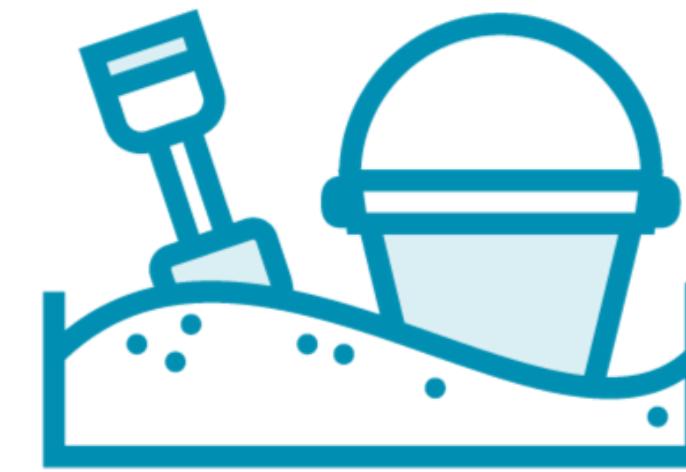
CLOUD ENGINEER

@ryanmurakami ryanlewis.dev

This Is the Course for You



Investigating CI/CD

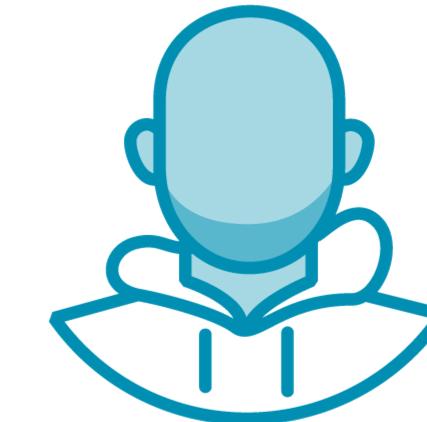


Just curious

Let Me Tell You a Story



Inside Hamster Ball Fantasy League



The New Hotness



HBFL

Races Hamsters Leaderboards Login

Hamster Ball Fantasy League

The ONLY place on the web to follow the Hamster Ball League

Featured Hamsters



Zepto

Rank:

Type: Speedball



Milkshake

Rank:

Type: Speedball



Fievel

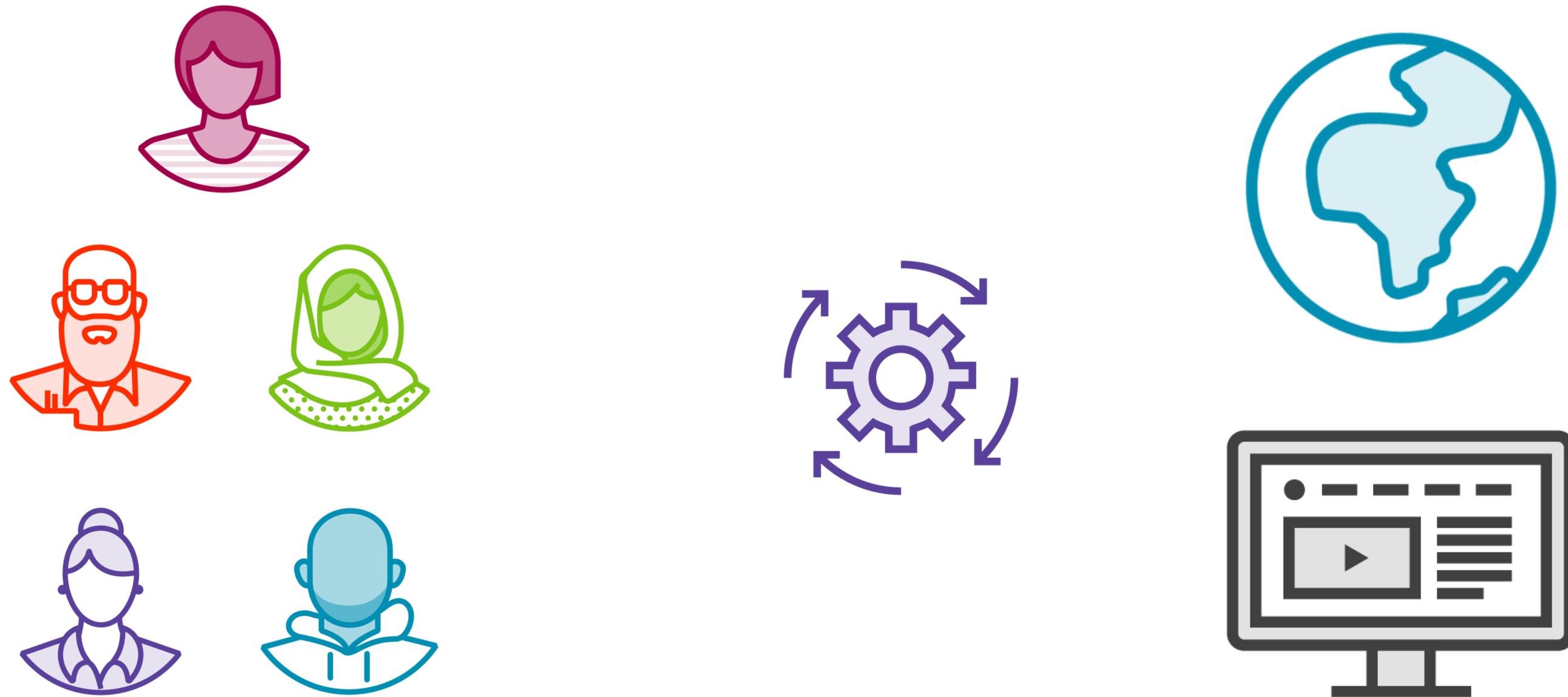
Rank:

Type: Tiny Terror

We've All Seen This Before



A Plan to Reach the Finish Line

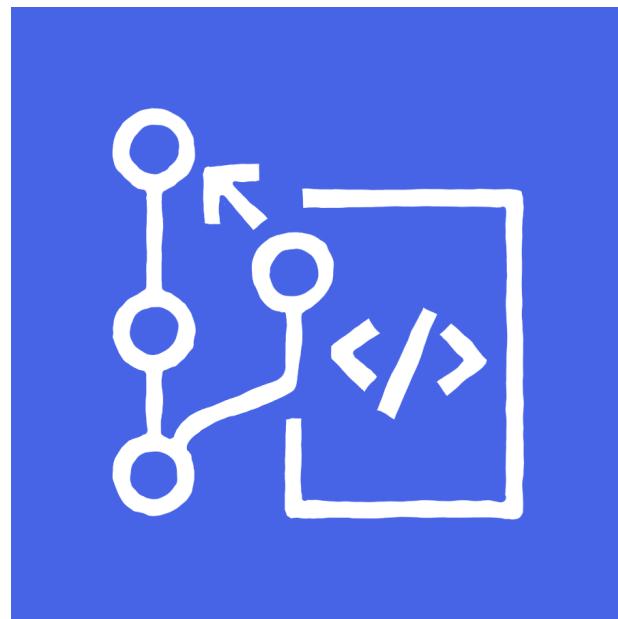


AWS CodePipeline

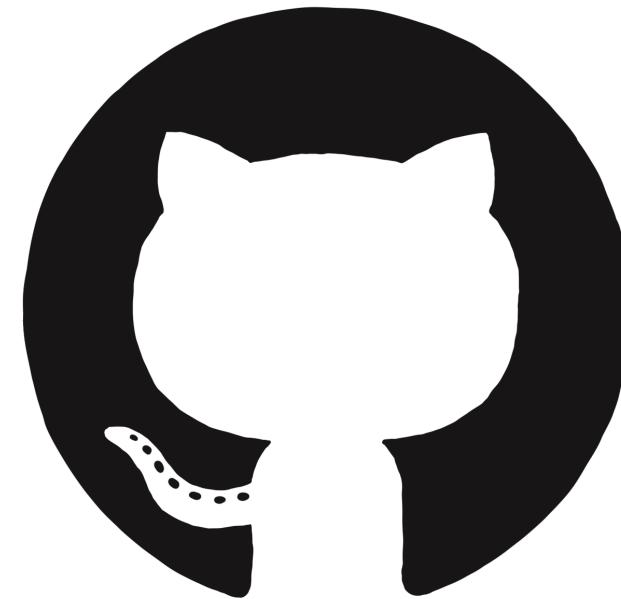


CodePipeline Integrated Services

Source Control



CodeCommit



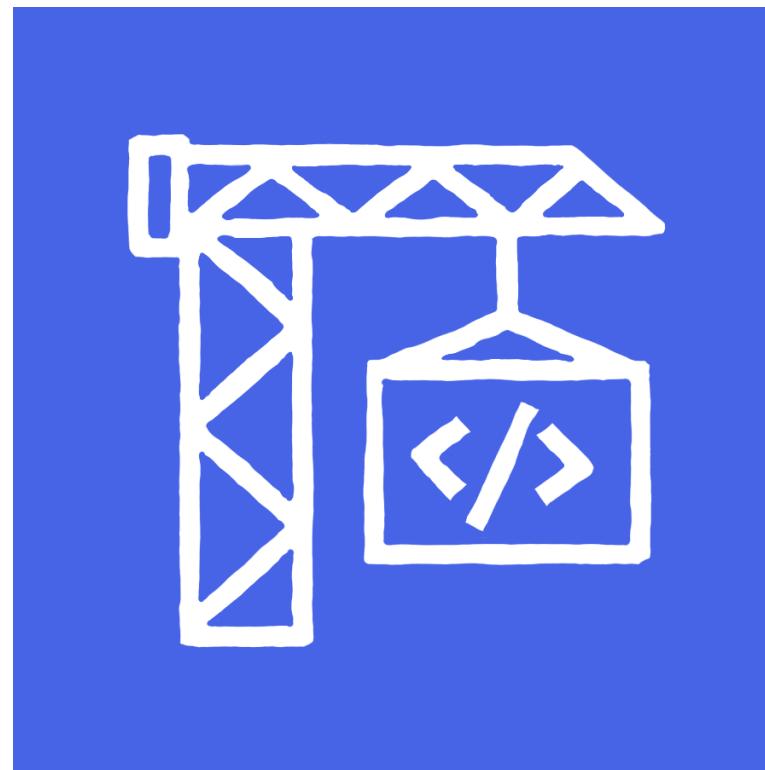
GitHub



Bitbucket

CodePipeline Integrated Services

Build



CodeBuild

CodePipeline Integrated Services

Lambda



Lambda

CodePipeline Integrated Services

Deployment



CodeDeploy



ECS



Elastic
Beanstalk



Fargate

Investigating Similar Services



Travis CI

Comparing CI/CD Options

Travis CI & CircleCI

Automated CI service

Workflow definition

Triggered from source control

Cloud provider agnostic

Can use AWS API to do most things

AWS CodePipeline

Automated CI service

Workflow definition

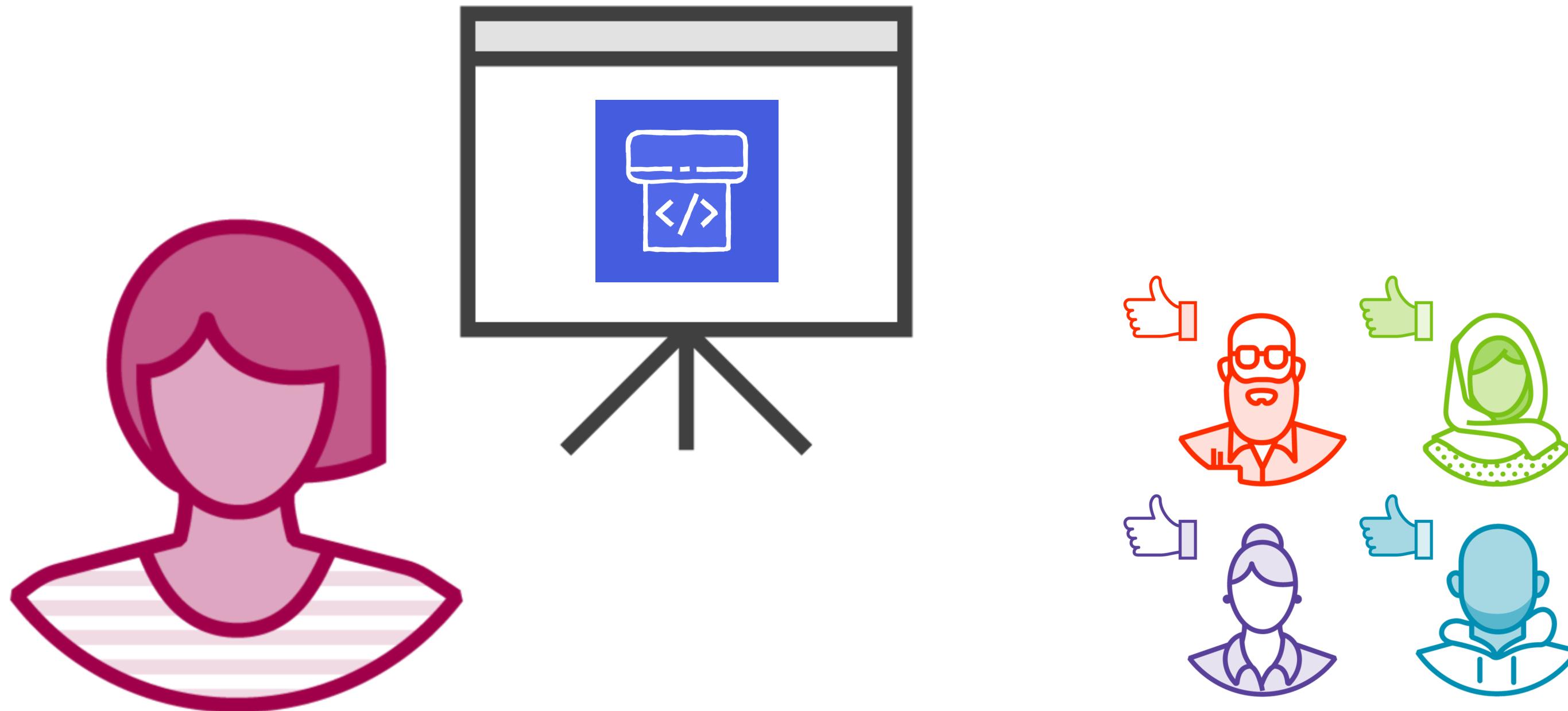
Triggered from source control

Low barrier to entry for AWS teams

\$1 per pipeline

Easy setup for AWS applications

How About CodePipeline?



Overview

A visual pipeline creation process

More stages, more fun

Running a test marathon

Please approve my pipeline paperwork

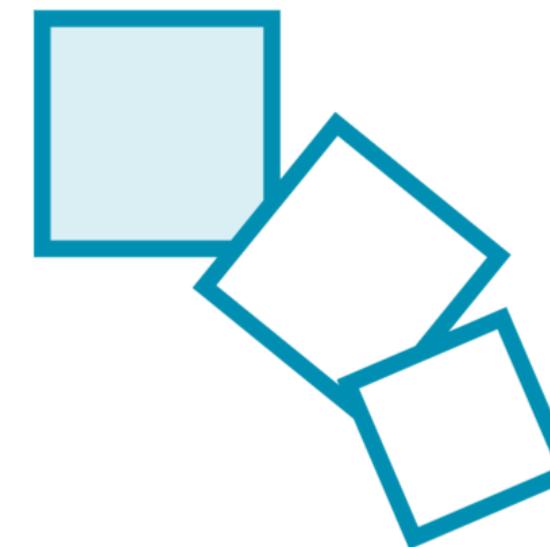
Copy that, CloudFormation

Module Overview

Advanced CodePipeline Practices



Lambda Invocations



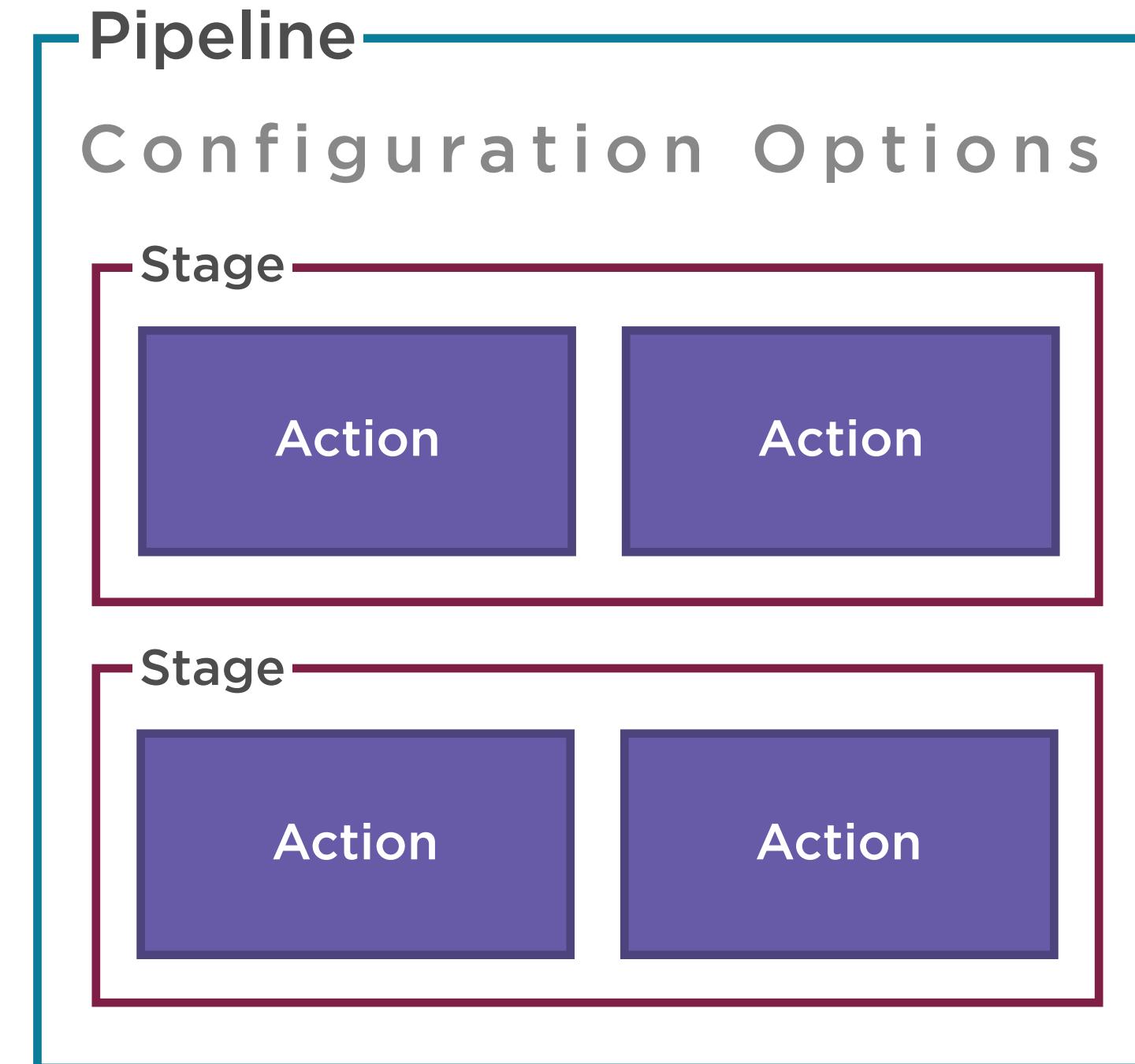
Stage Transitions



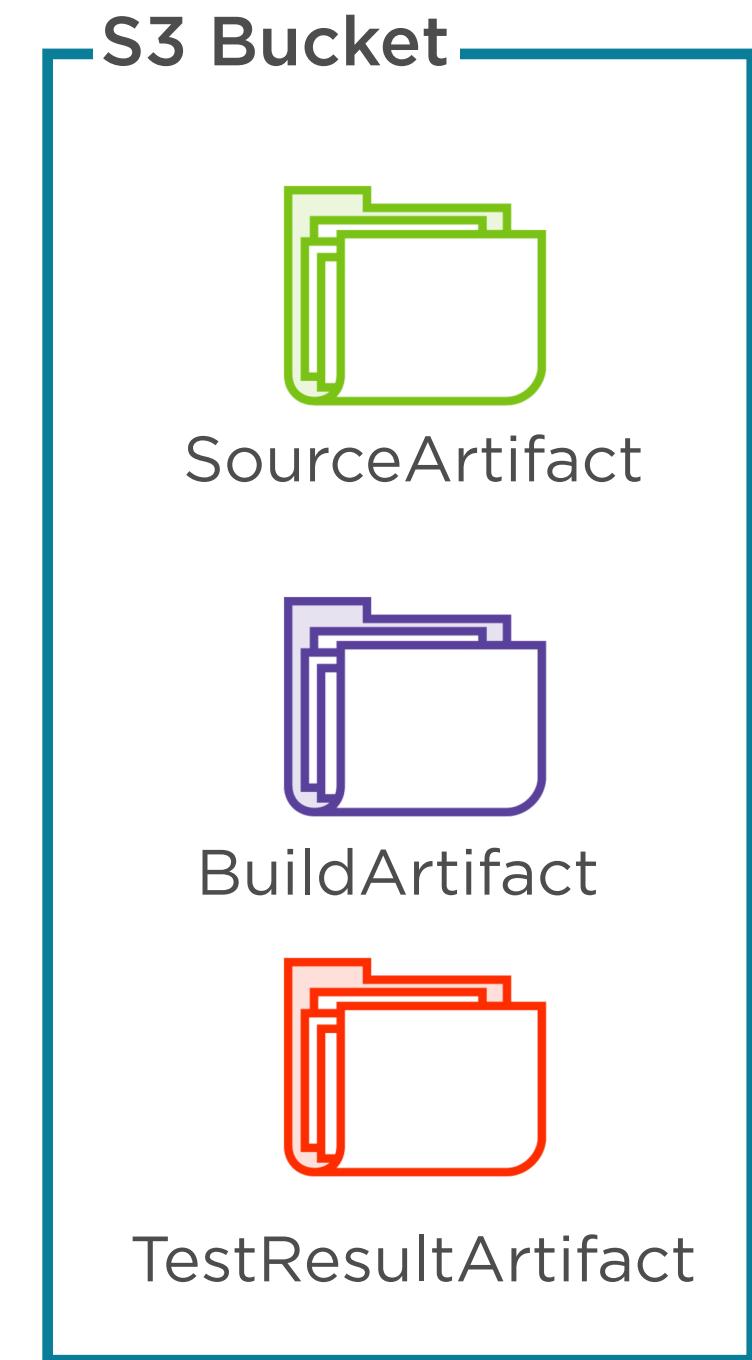
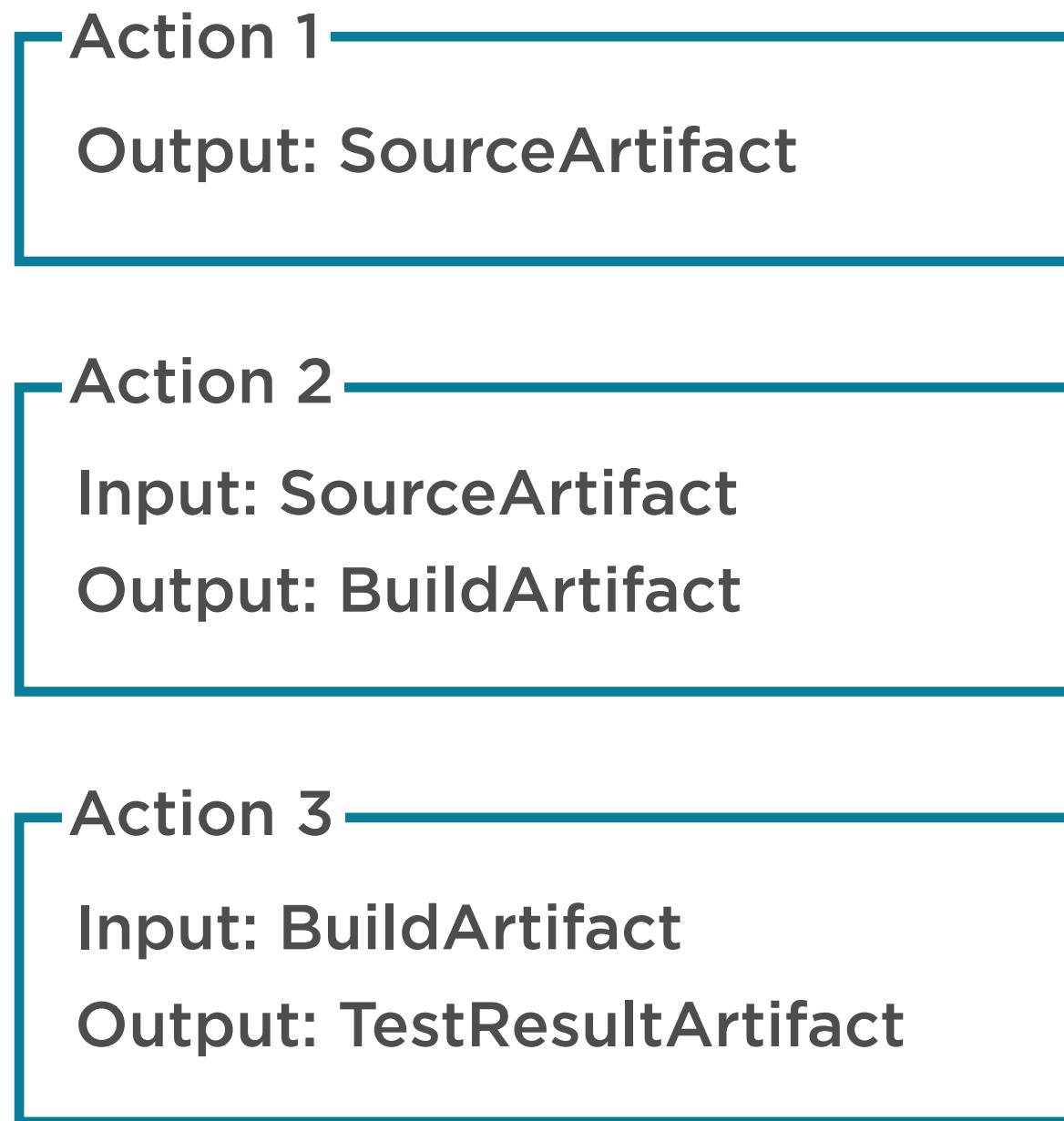
Monitoring and
Notifications

A Bird's Eye View of CodePipeline

CodePipeline Structure



State in CodePipeline



Pipeline Configuration Options



Pipeline name identifier



Service role with access to most AWS services



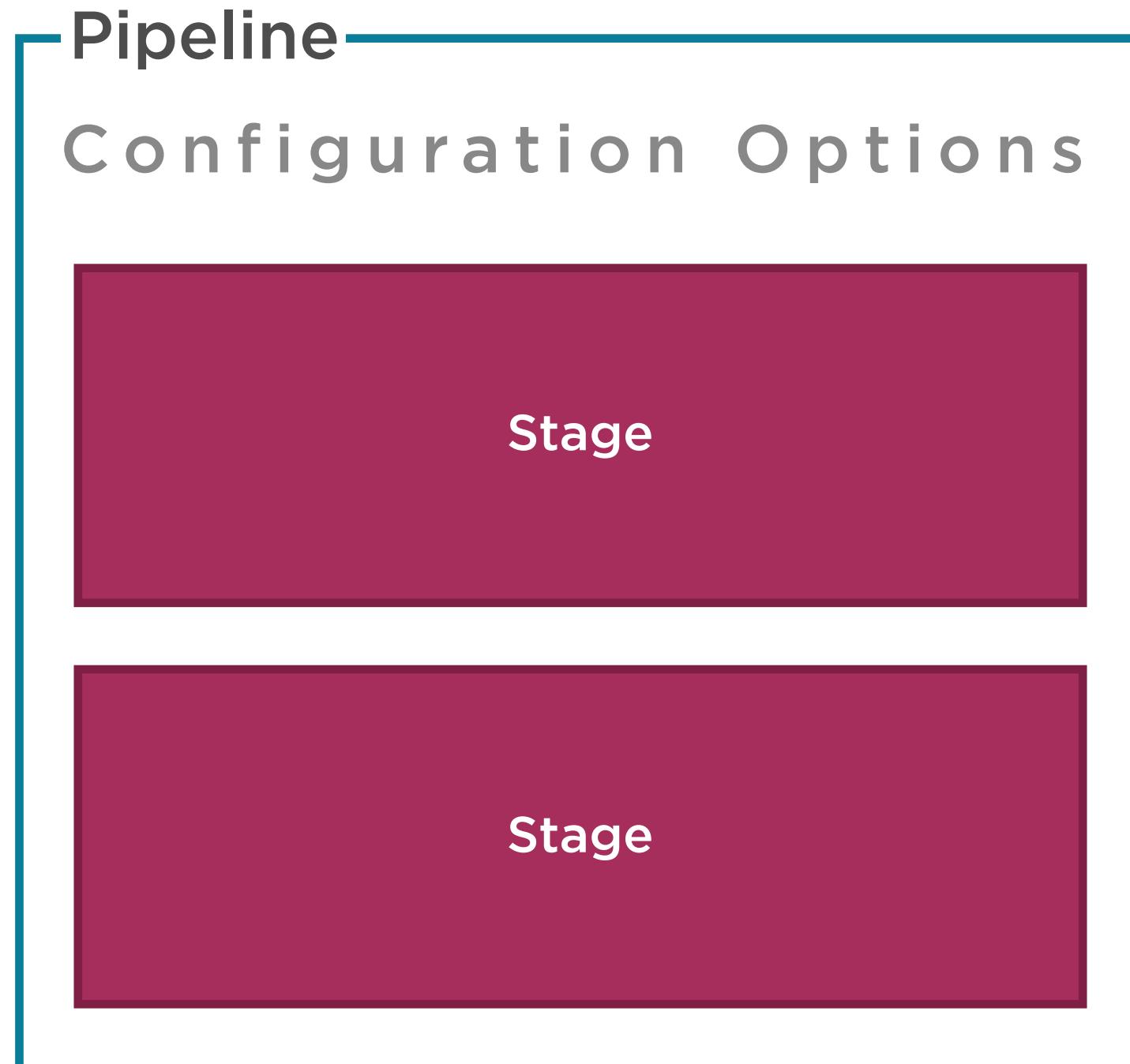
S3 bucket to act as the artifact store



Encryption key: Yours in KMS or default Amazon-provided

Stages in a Pipeline

Stages
Min: 2
Max: 10

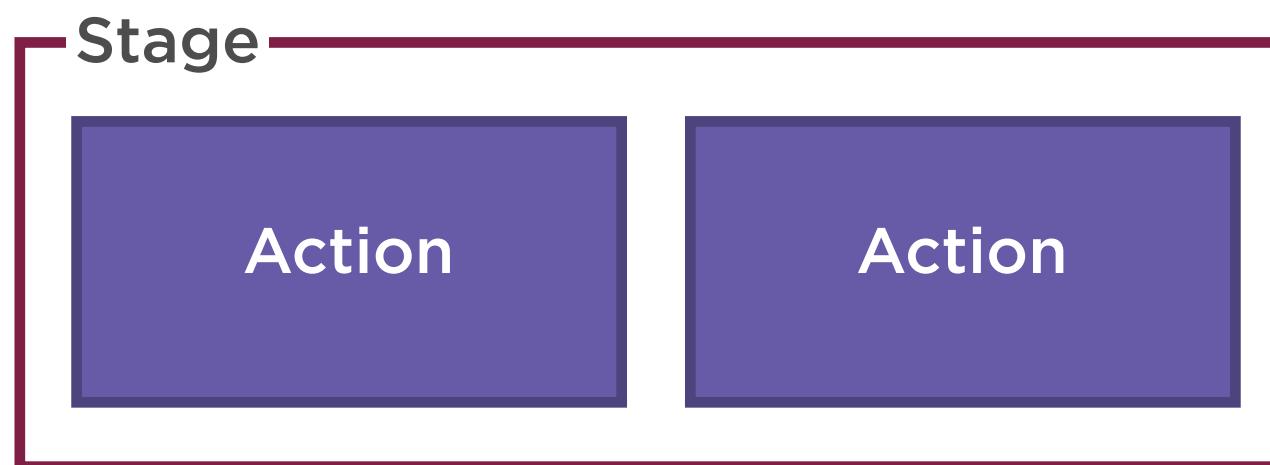
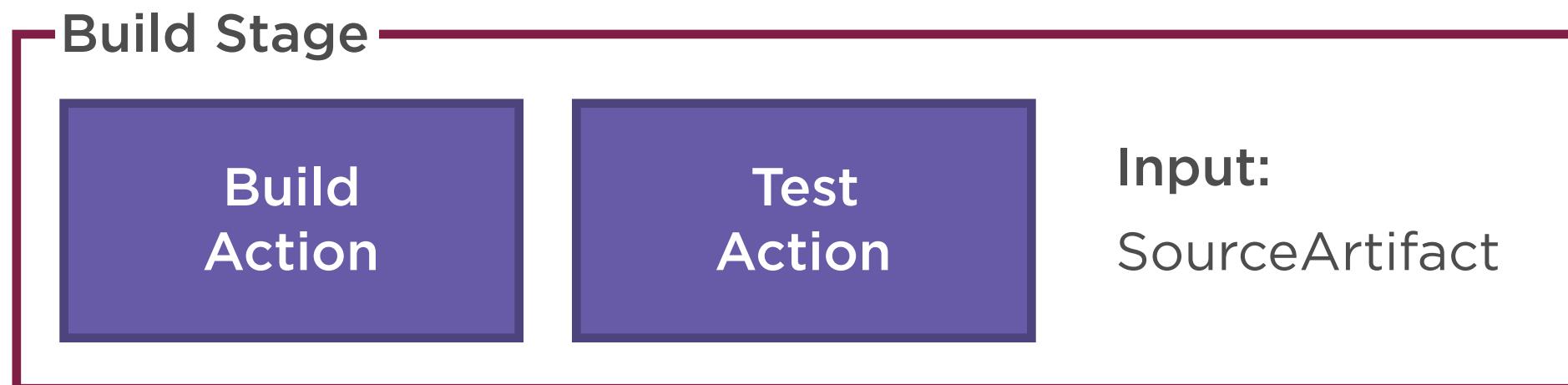
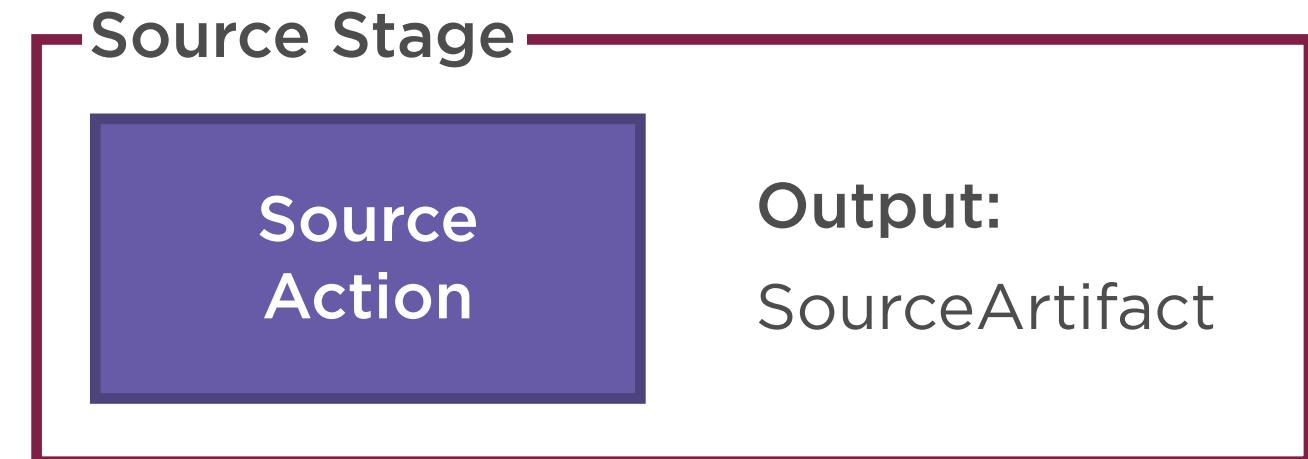


Anatomy of a Stage

First Stage:

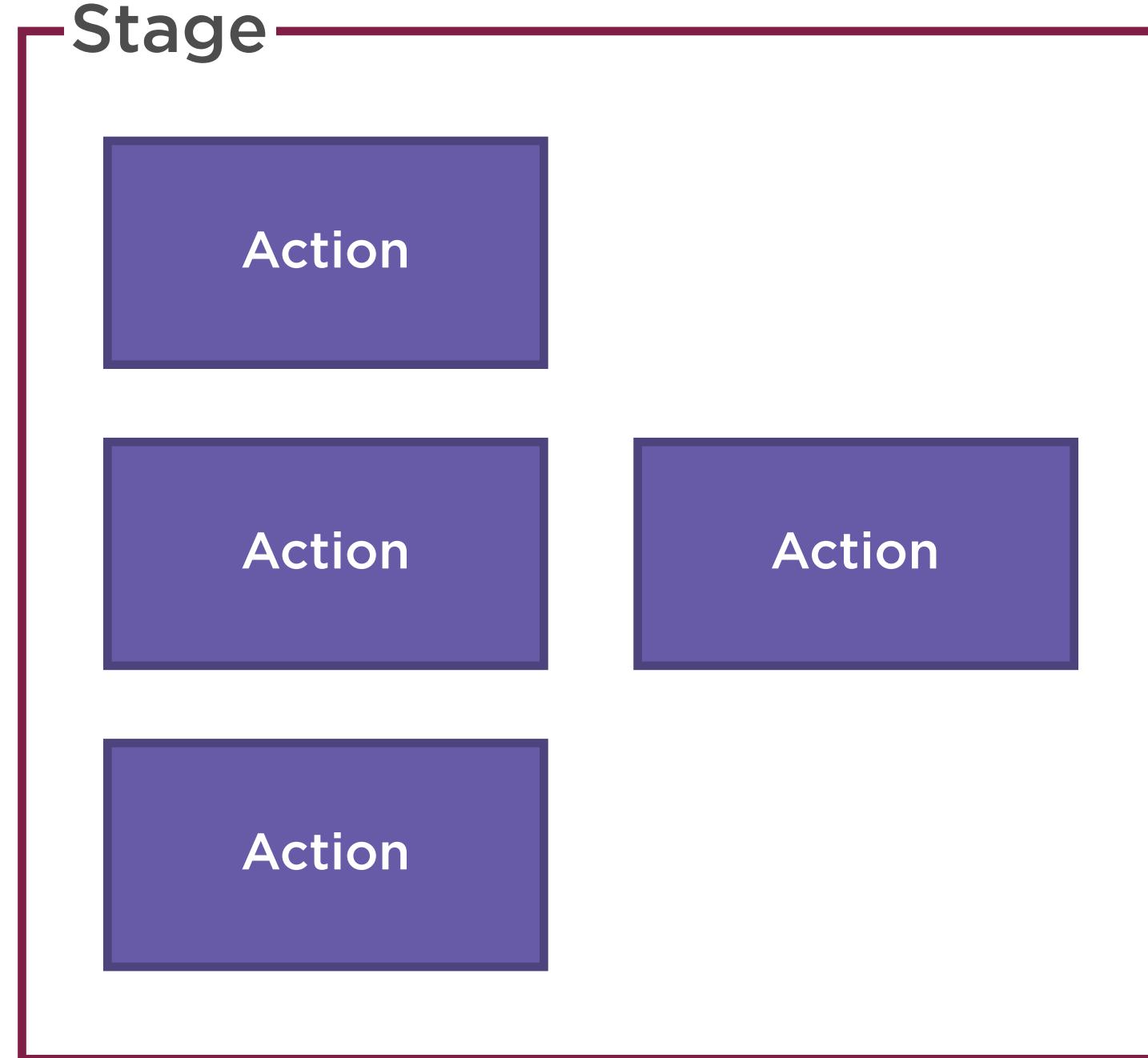
Must contain one source action

Cannot contain any other action type



Actions in a Stage

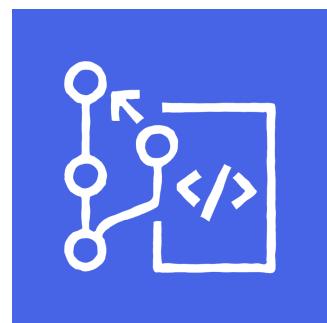
Actions
Min: 1
Max: 50



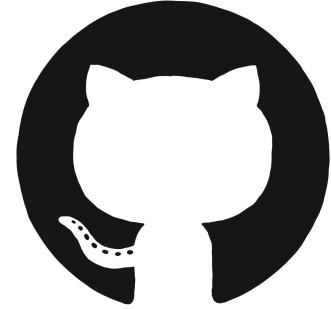
Can only appear in the first stage

Pulls in source code

Source action providers:



CodeCommit



GitHub

Bitbucket



Bitbucket



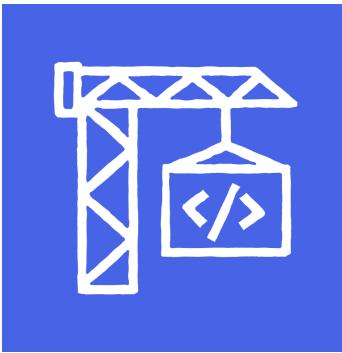
ECR

S3

Build action providers:



Jenkins

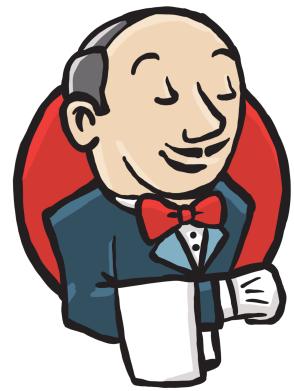


CodeBuild

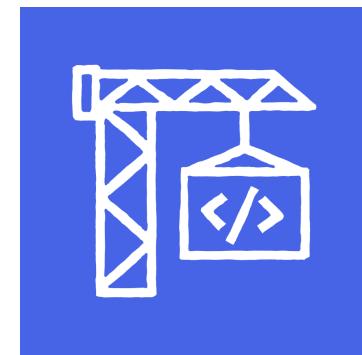
Examples: Build application, compile code

Can run basically any commands you want

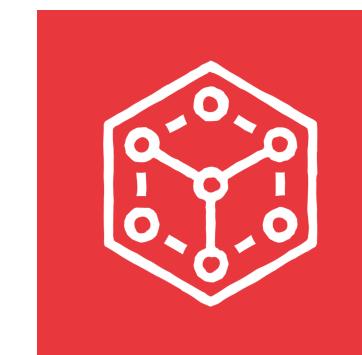
Test action providers:



Jenkins



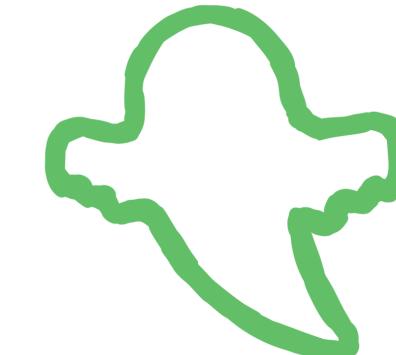
CodeBuild



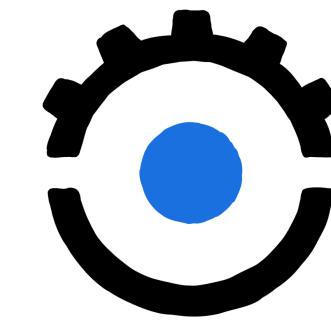
Device Farm



BlazeMeter



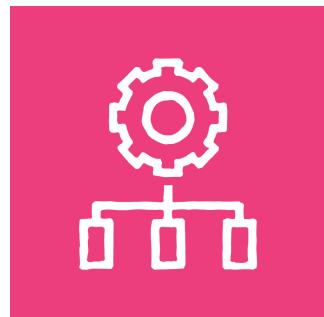
Ghost Inspector



Runscope

Deploy built artifacts from pipeline

Deploy action providers:



OpsWorks



ECS



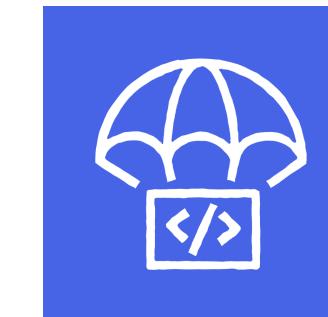
CloudFormation



S3



Elastic
Beanstalk



CodeDeploy



Service
Catalog

CodePipeline Action Type

Approval

Manually approval required to continue execution

CodePipeline Action Type

Invoke

Invoke an AWS Lambda function

Best Way to Create a Pipeline

Use the AWS console to create a pipeline

**Use a CloudFormation template to build
a pipeline**

Why Use CloudFormation for CodePipeline?

- Keep infrastructure in source control**
- Easy to recreate pipeline**
- Easy to change pipelines and create new**

Preparing for a Pipeline

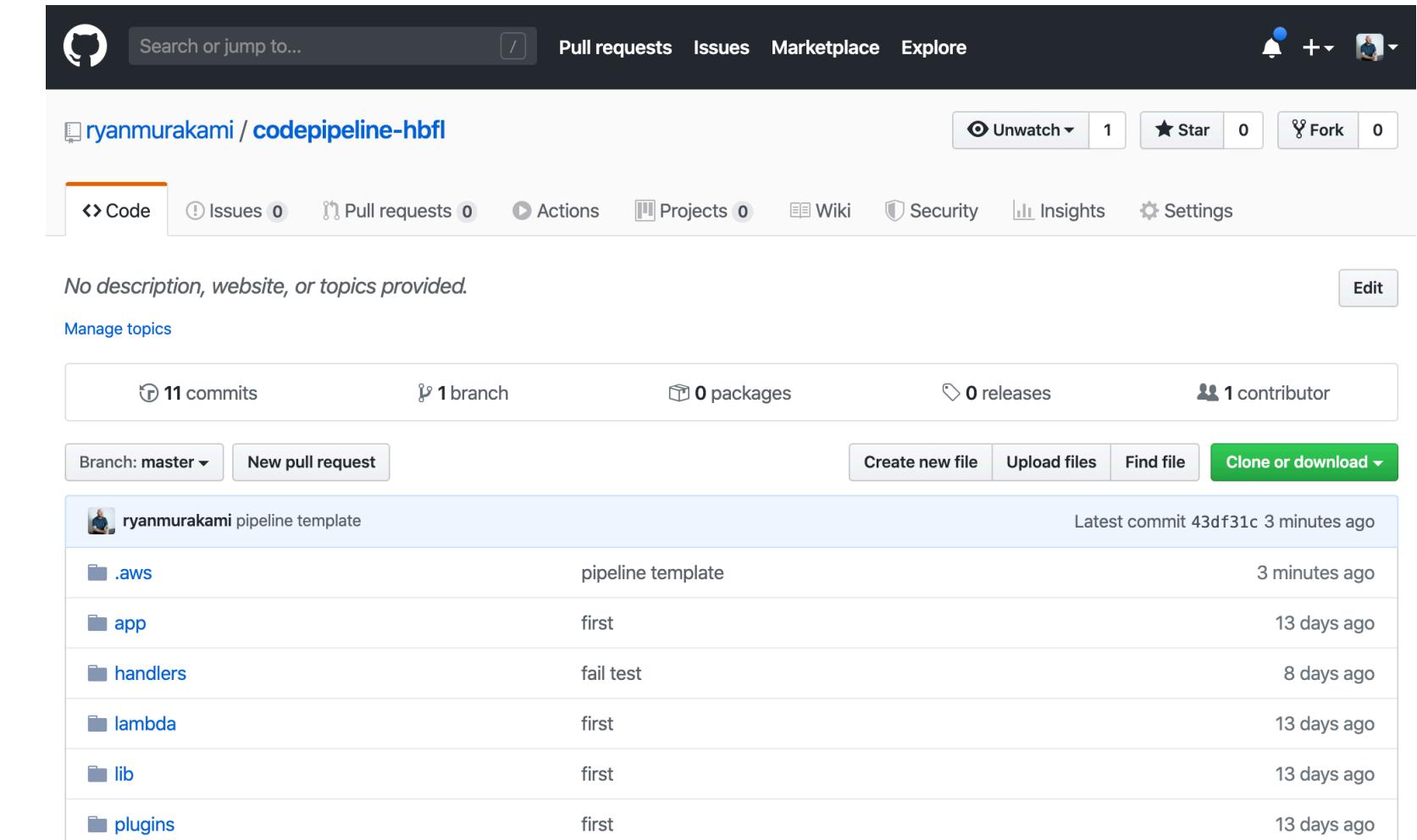
Demo

Create repository in CodeCommit

**Create application and environment in
Elastic Beanstalk**

Course Demo Project Code

Or download
on Pluralsight
under “Exercise
Files”



github.com/ryanmurakami/codepipeline-hbfl

Create a non-root user in
your AWS account

Install Git

The screenshot shows the official Git website at <https://git-scm.com>. The page features a large header with the Git logo and the tagline "git --distributed-is-the-new-centralized". Below the header, there's a brief introduction to Git as a free and open source distributed version control system. To the right of the text is a 3D-style diagram illustrating a distributed network of repositories connected by red lines. The main content area is divided into several sections: "About" (with a gear icon), "Downloads" (with a download arrow icon), "Documentation" (with a book icon), and "Community" (with a speech bubble icon). A sidebar on the right displays the "Latest source Release" (2.25.1) with a "Release Notes (2020-02-17)" link and a "Download 2.23.0 for Mac" button.

git --distributed-is-the-new-centralized

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

About
The advantages of Git compared to other source control systems.

Downloads
GUI clients and binary releases for all major platforms

Documentation
Command reference pages, Pro Git book content, videos and other material.

Community
Get involved! Bug reporting, mailing list, chat, development

Latest source Release
2.25.1
Release Notes (2020-02-17)
Download 2.23.0 for Mac

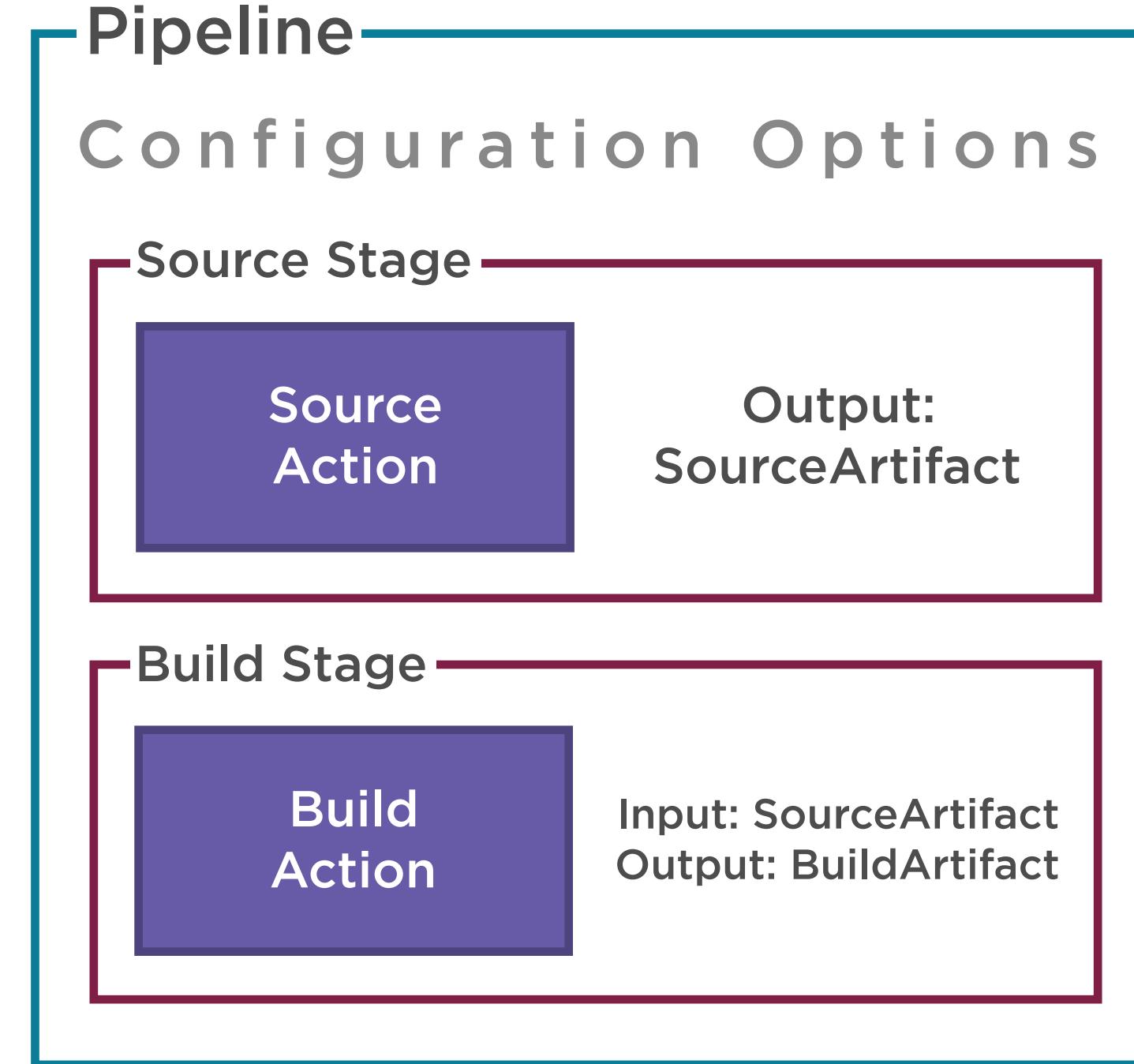
<https://git-scm.com>

Creating a Pipeline

Demo

Build a two-stage pipeline in CodePipeline

Two-stage Demo Pipeline



What the Pipeline Will Build



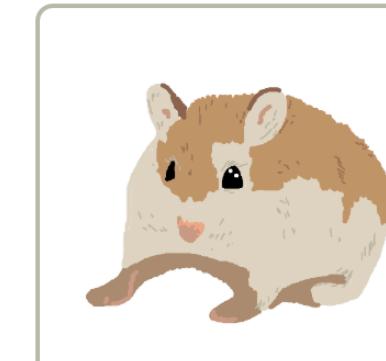
Featured Hamsters



Zepto

Rank:

Type: Speedball



Milkshake

Rank:

Type: Speedball



Fievel

Rank:

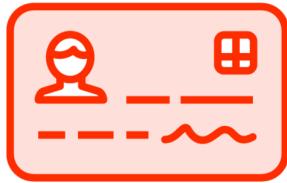
Type: Tiny Terror

Deploying from CodePipeline

Demo

Add a deploy stage and action to the demo pipeline

Pipeline Execution in Detail



Each execution has a unique ID



Executions run until they succeed, fail, or are stopped

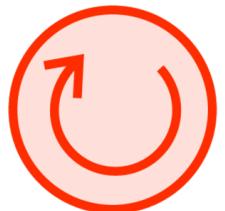


Pipelines can run multiple executions; stages only one at a time

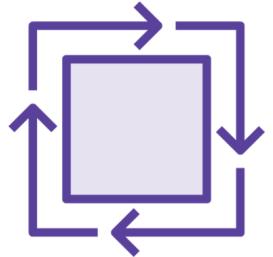


Stages lock when executing; subsequent executions must wait

More Pipeline Execution in Detail



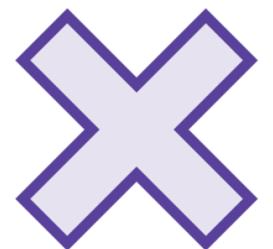
A failed pipeline execution can be retried



Executions can supersede each other



If an execution is waiting at a locked stage, it can be superseded



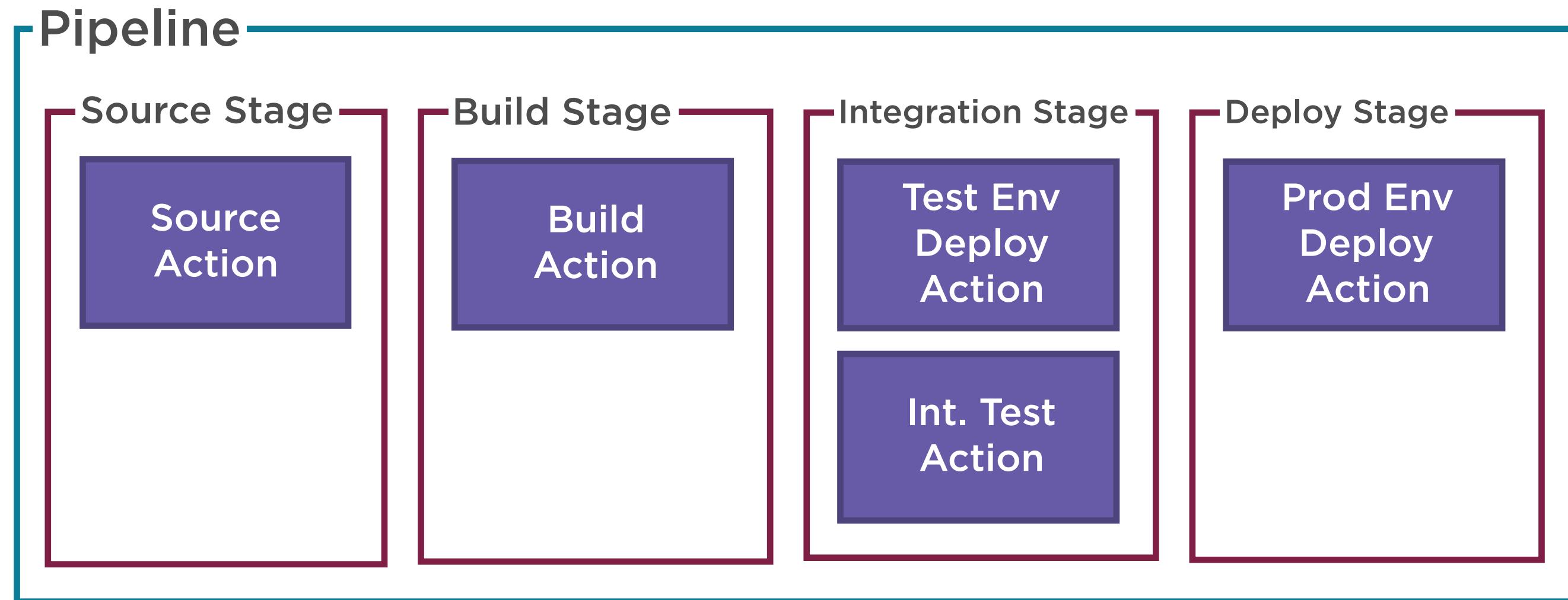
A superseded execution will not continue and cannot be retried

Running Tests in CodePipeline

Demo

Add a test action before our deploy action

Integration Test Pipeline Example



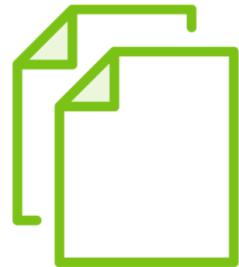
Manual Approvals in CodePipeline

Demo

Add a manual approval action

Using CodePipeline with CloudFormation

Reasons to Use CloudFormation



Easy to reproduce infrastructure in other regions, accounts



Can track infrastructure changes in source control



Quick and easy removal of resources

Demo

Use CloudFormation template to recreate pipeline

Summary

A pipeline tale for the ages

All the perks in a pipeline

Make way for the ye royal pipeline

Lights, pipeline, action!

A proper pipeline approval process

Let CloudFormation do all the hard work

Up Next:

Advanced CodePipeline Practices
