

CICD for PHP on AWS

Longhorn PHP 2018

Michael Moussa

@michaelmoussa

<https://joind.in/talk/e033f>



Who am I?

Who am I?

- Product Architect at Rackspace



Who am I?

- Product Architect at Rackspace
- 18 year PHP developer &
DevOps engineer



Who am I?

- Product Architect at Rackspace
- 18 year PHP developer & DevOps engineer
- AWS Certified Solutions Architect



Objective

Objective

Set up CICD for a non-trivial PHP application
running on AWS, and deploy it with zero downtime.

CICD?

Continuous Integration

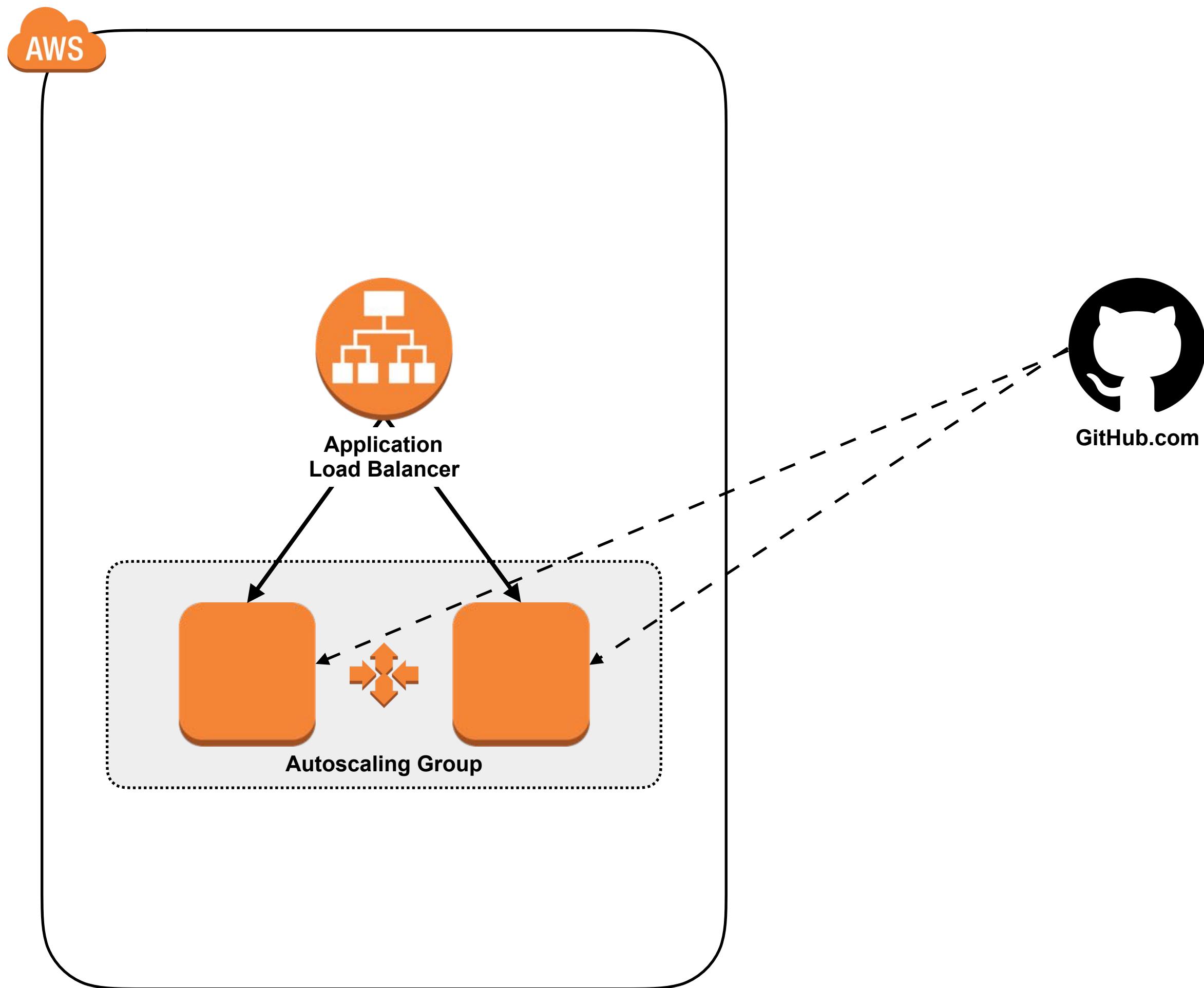
CICD?

Continuous Deployment

CICD?

Continuous Delivery

Getting started



CodeDeploy



CodeDeploy

- Automates application deployments



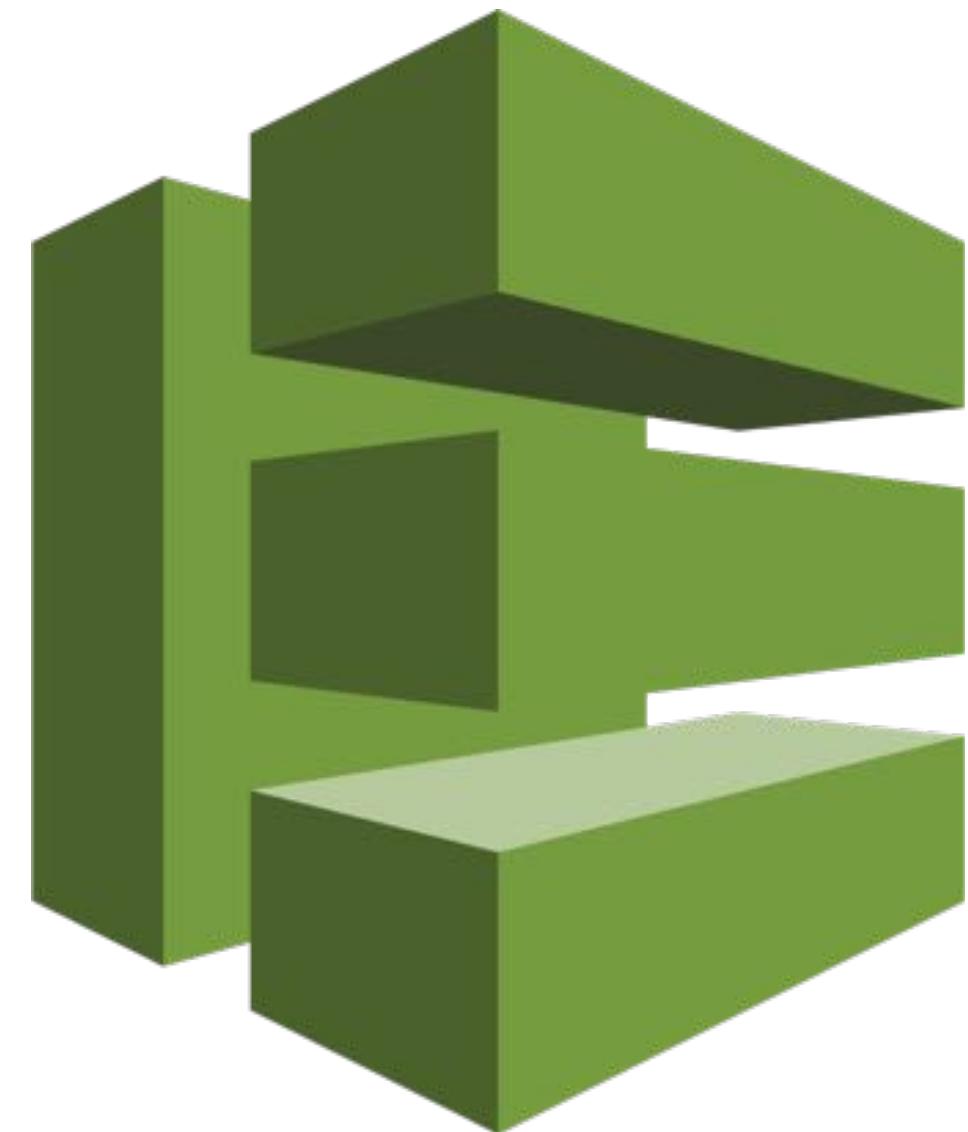
CodeDeploy

- Automates application deployments
 - EC2 instances
 - AWS Lambda
 - On-premise servers



CodeDeploy

- Automates application deployments
 - EC2 instances
 - AWS Lambda
 - On-premise servers
- Deploy from GitHub or Amazon S3



CodeDeploy

- Automates application deployments
 - EC2 instances
 - AWS Lambda
 - On-premise servers
- Deploy from GitHub or Amazon S3
- Highly customizable



CodeDeploy - Example



Assumptions

- "Hello, world!" application code hosted on GitHub

CodeDeploy - Example



Assumptions

- "Hello, world!" application code hosted on GitHub
- Autoscaling group (3 instances) with application running

CodeDeploy - Example



Assumptions

- "Hello, world!" application code hosted on GitHub
- Autoscaling group (3 instances) with application running
- First time setup steps completed:
 - <https://docs.aws.amazon.com/codedeploy/latest/userguide/getting-started-codedeploy.html>

CodeDeploy - Get started



Sample deployment

Recommended for new AWS CodeDeploy users.



Custom deployment

Recommended if you already have instances and an application to deploy.

CodeDeploy - Create application



Application name*	DemoApplication
Compute Platform*	EC2/On-premises
Deployment group name*	DemoApplication



CodeDeploy - Deployment type



In-place deployment

Updates the instances in the deployment group with the latest application revision. During a deployment, each instance will be briefly taken offline for its update.



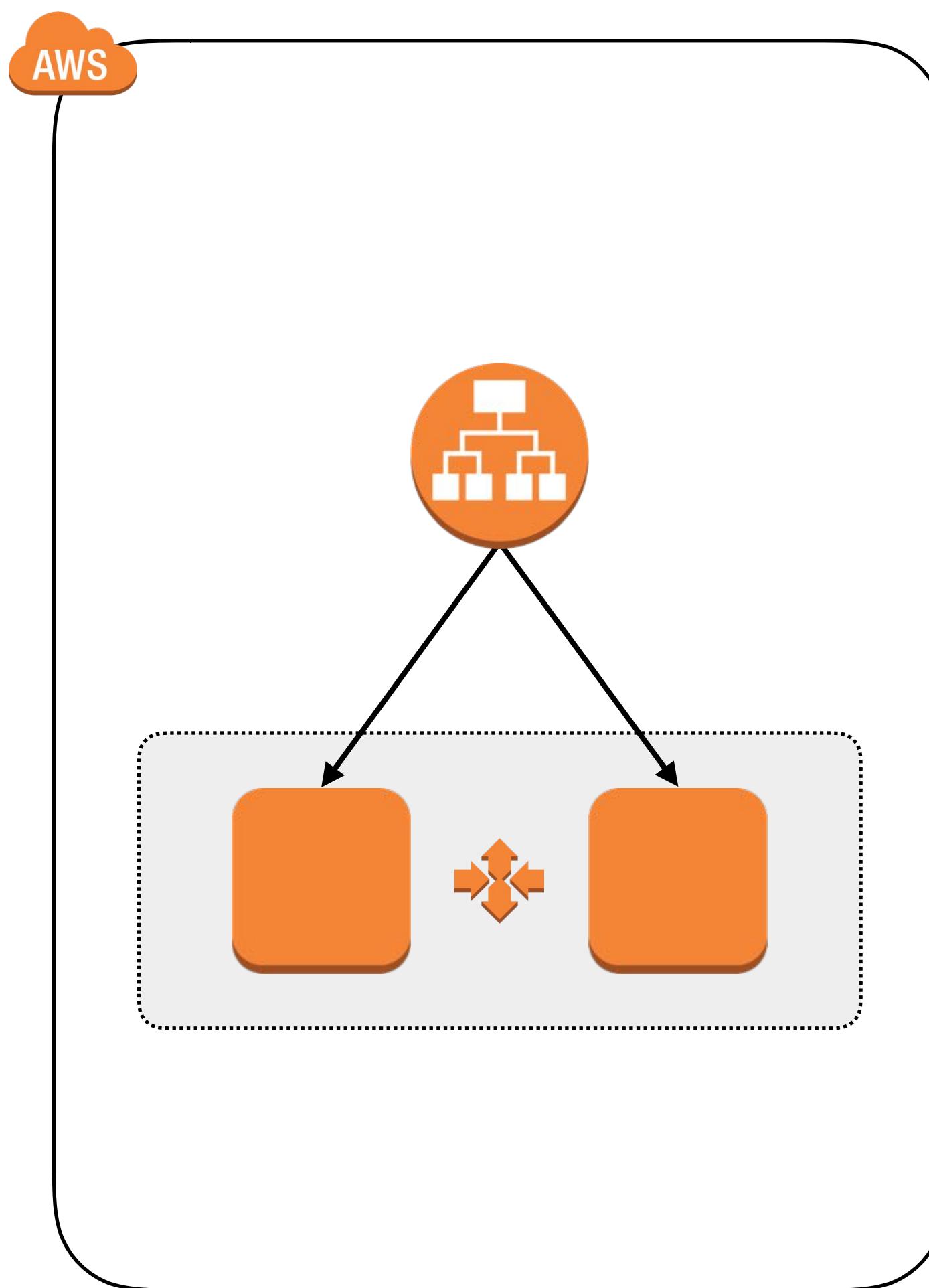
Blue/green deployment

Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

CodeDeploy - Deployment type



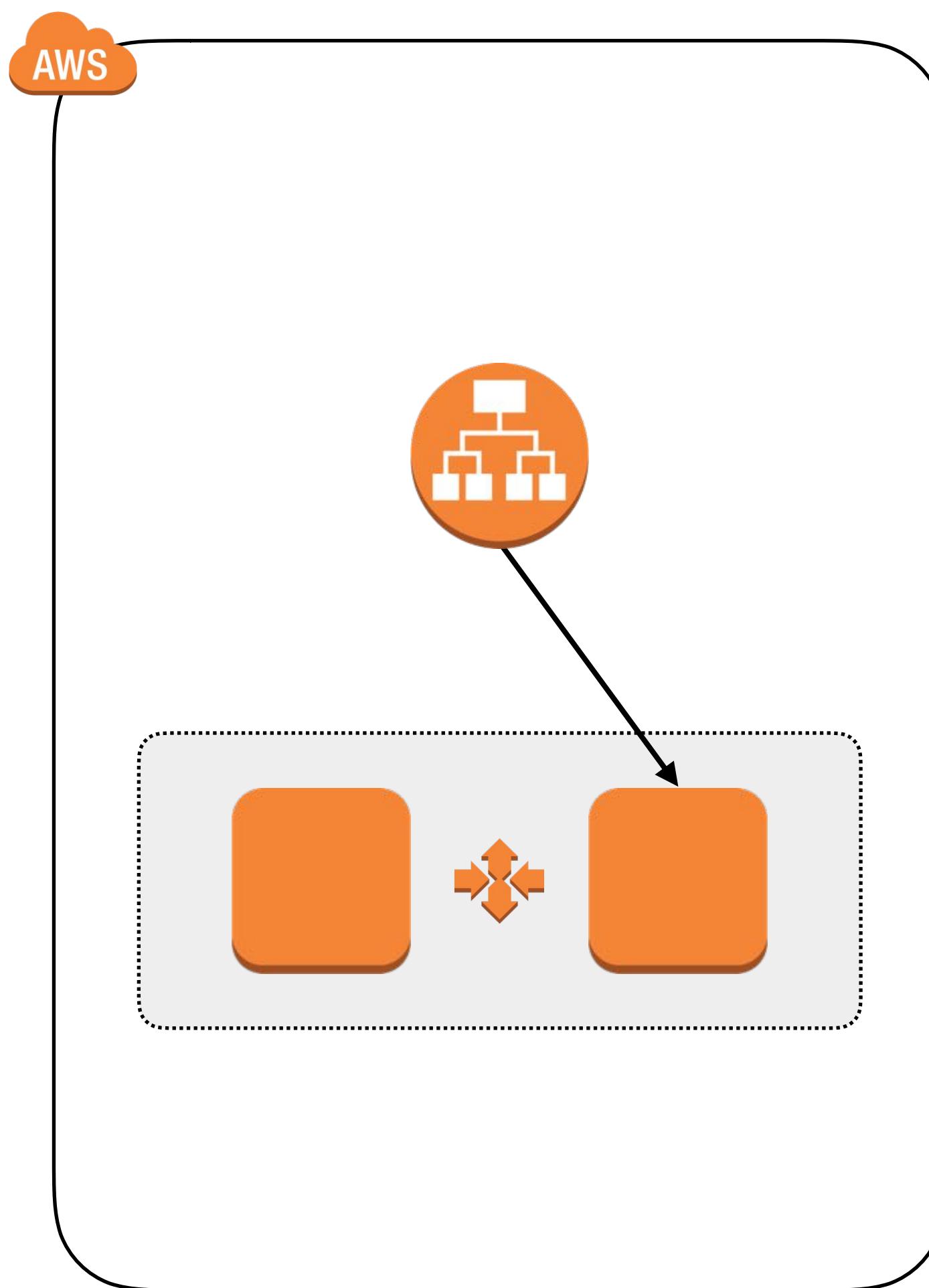
- In-place deployment
- Blue/green deployment



CodeDeploy - Deployment type



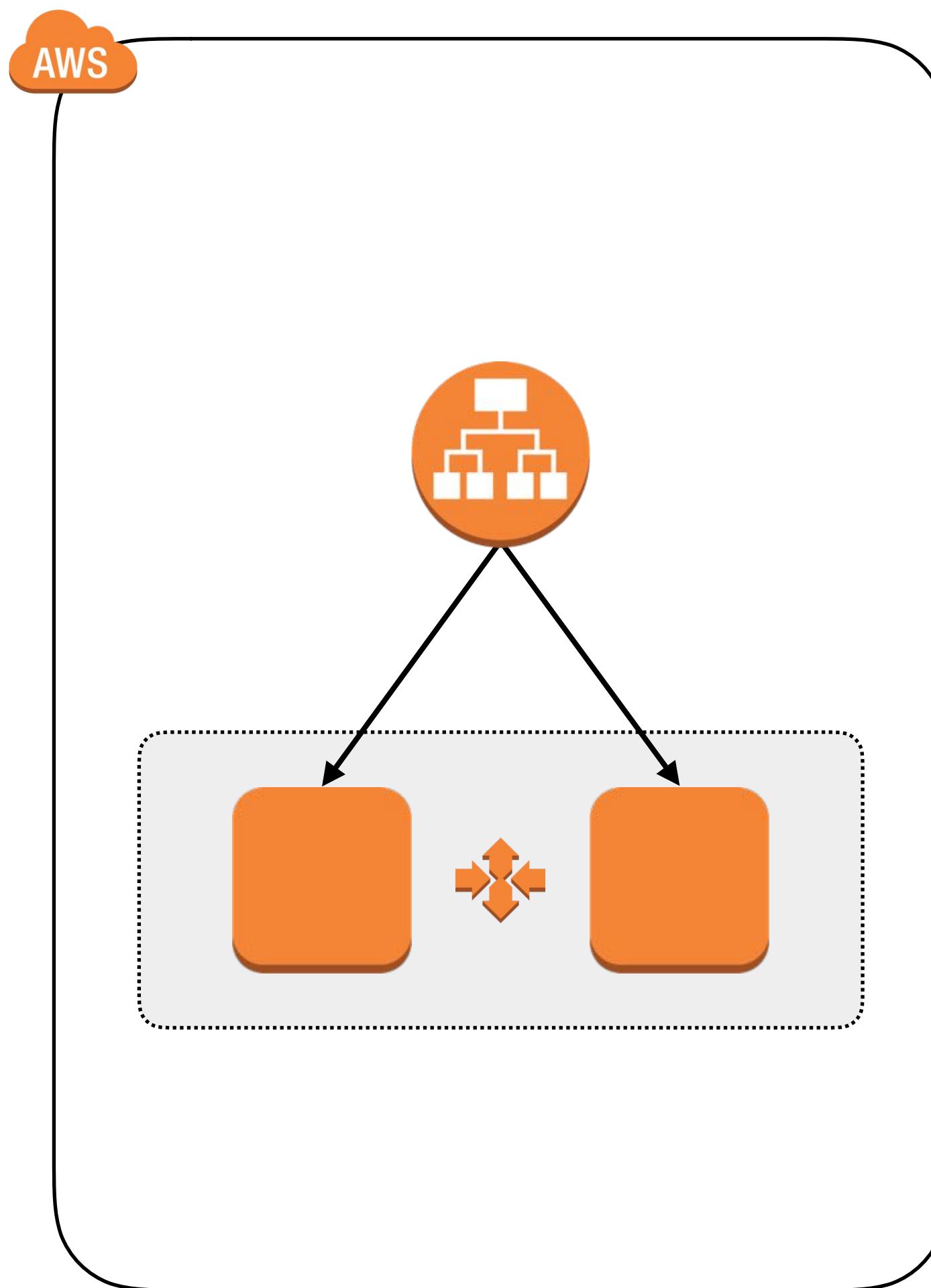
- In-place deployment
- Blue/green deployment



CodeDeploy - Deployment type



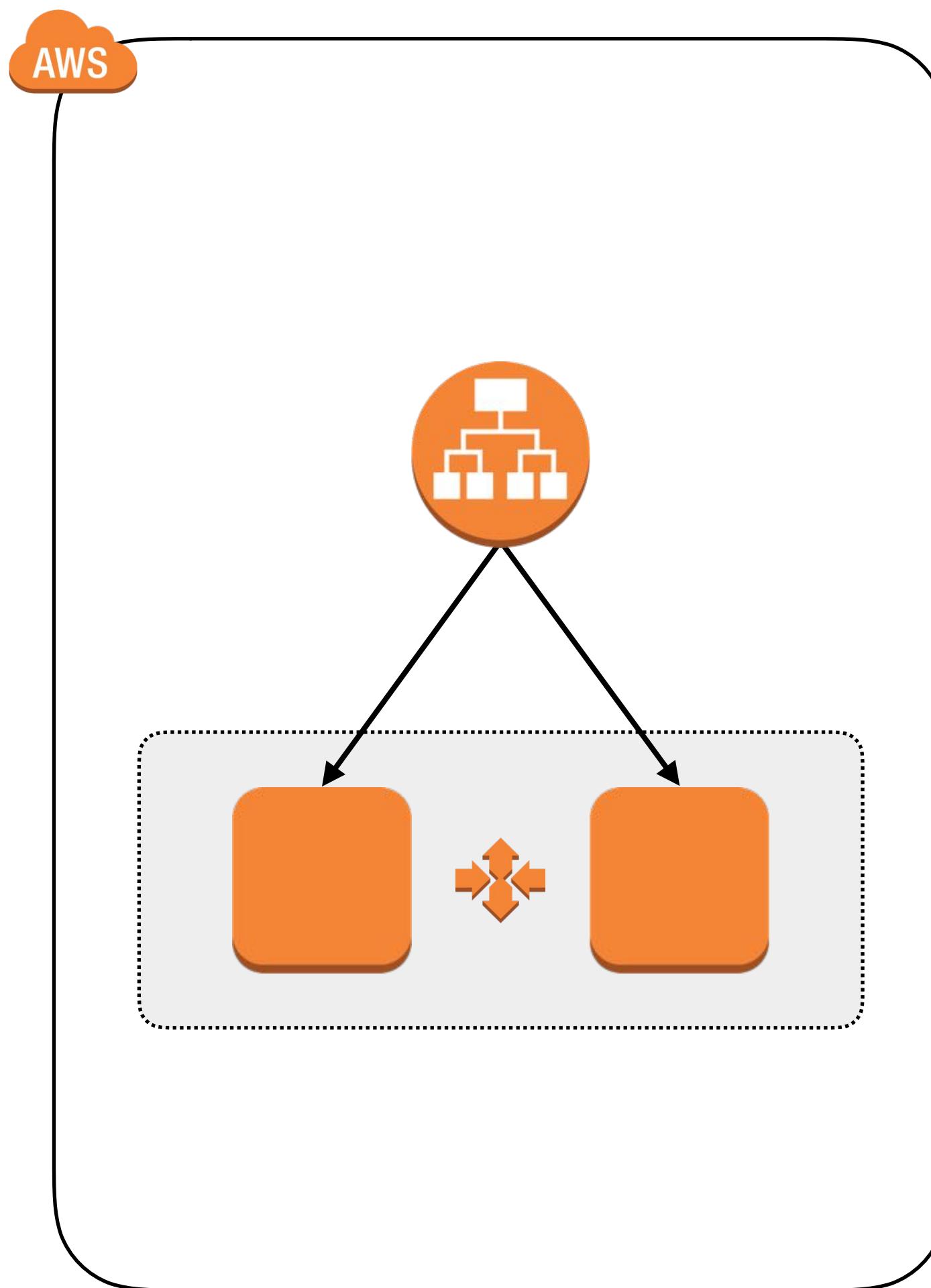
- In-place deployment
- Blue/green deployment



CodeDeploy - Deployment type



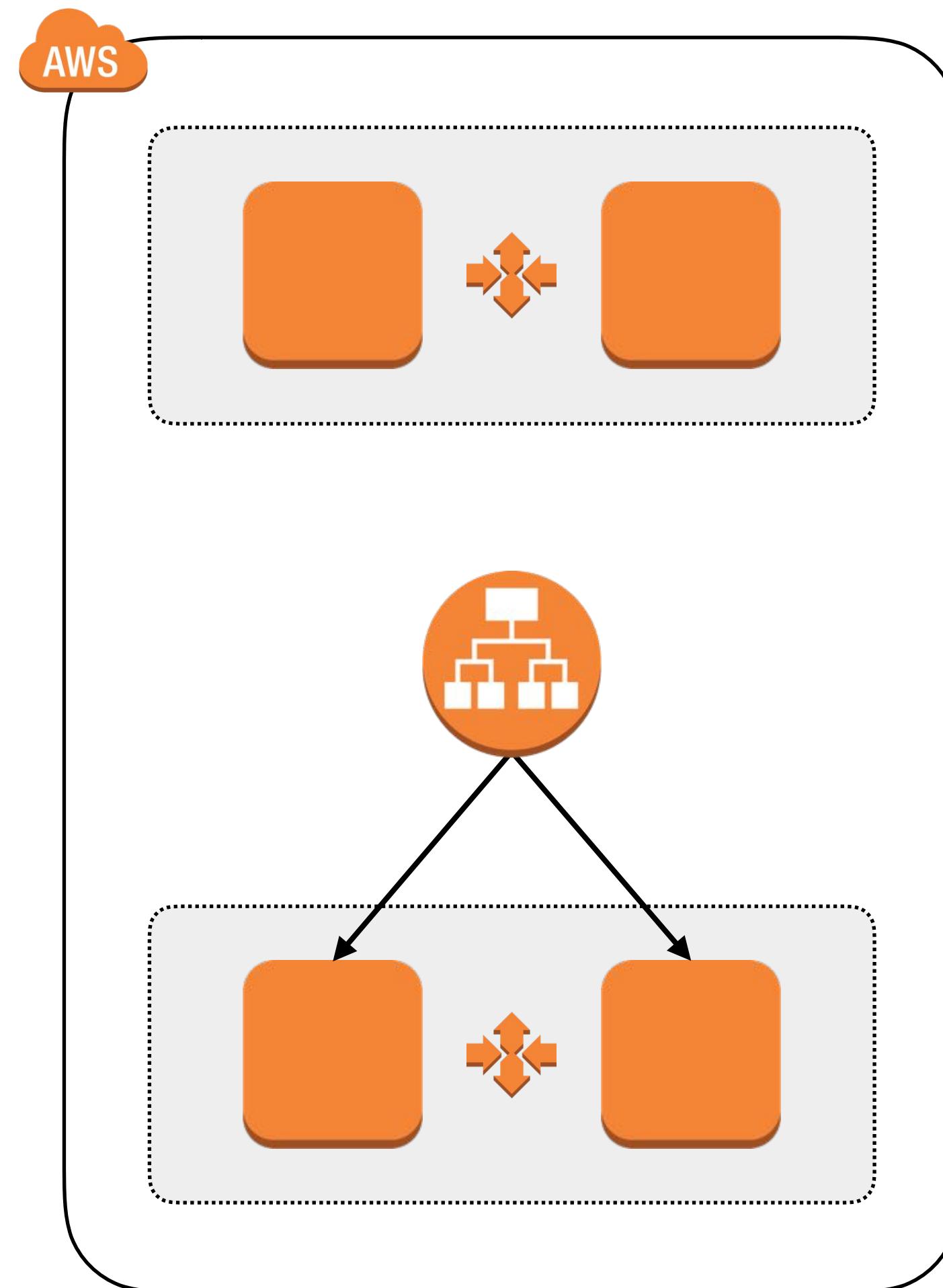
- In-place deployment
- **Blue/green deployment**



CodeDeploy - Deployment type



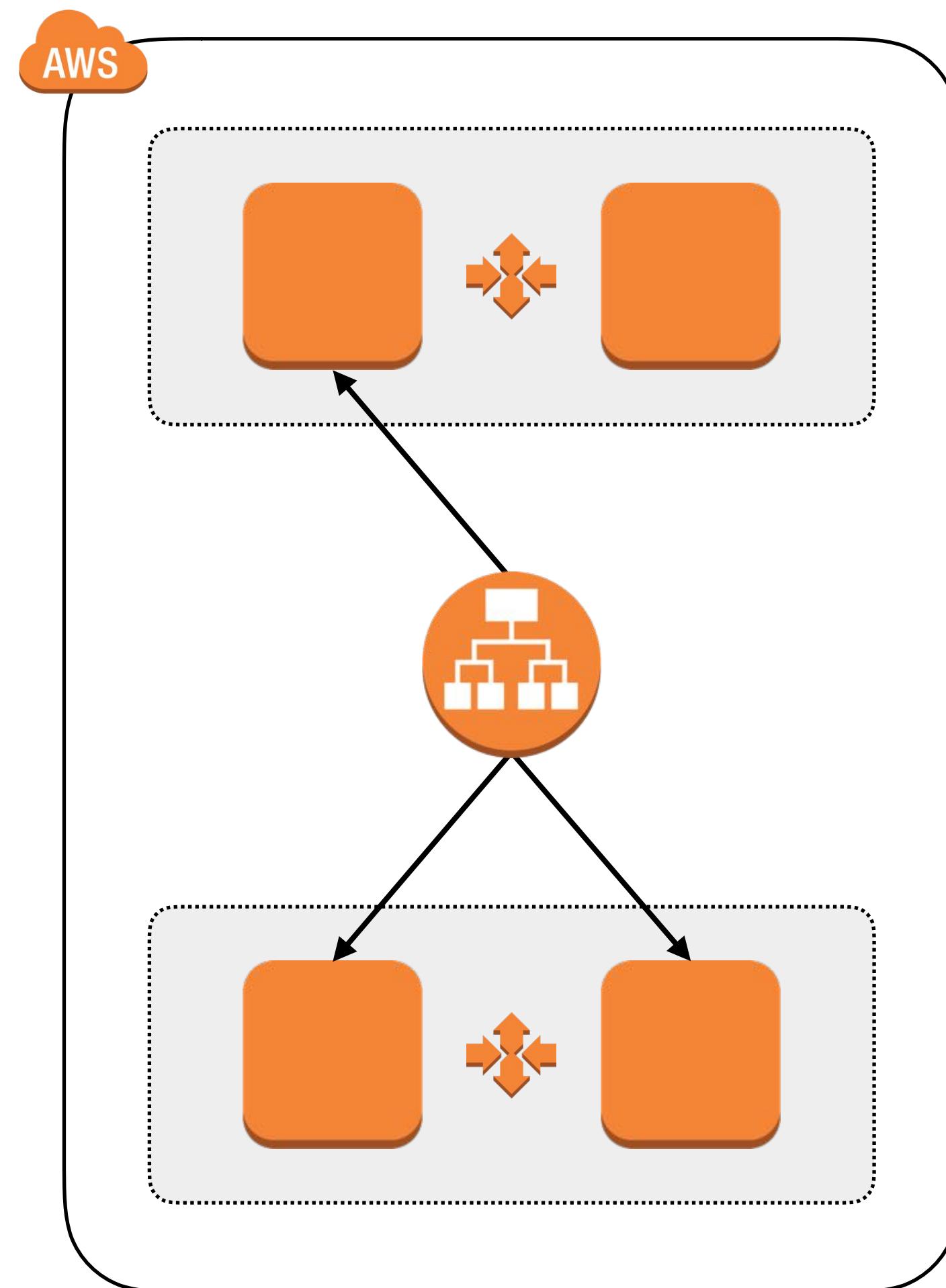
- In-place deployment
- **Blue/green deployment**



CodeDeploy - Deployment type



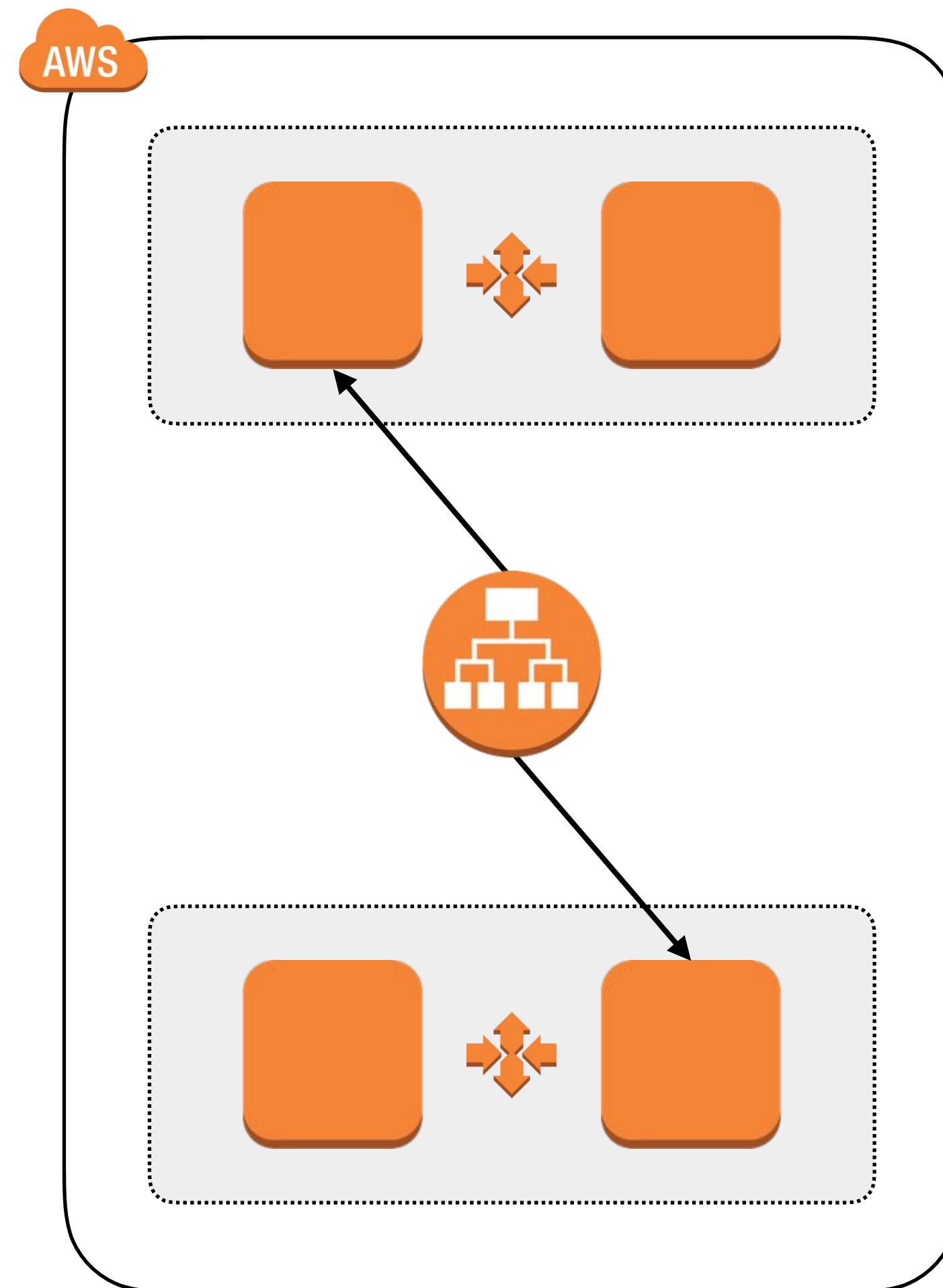
- In-place deployment
- **Blue/green deployment**



CodeDeploy - Deployment type



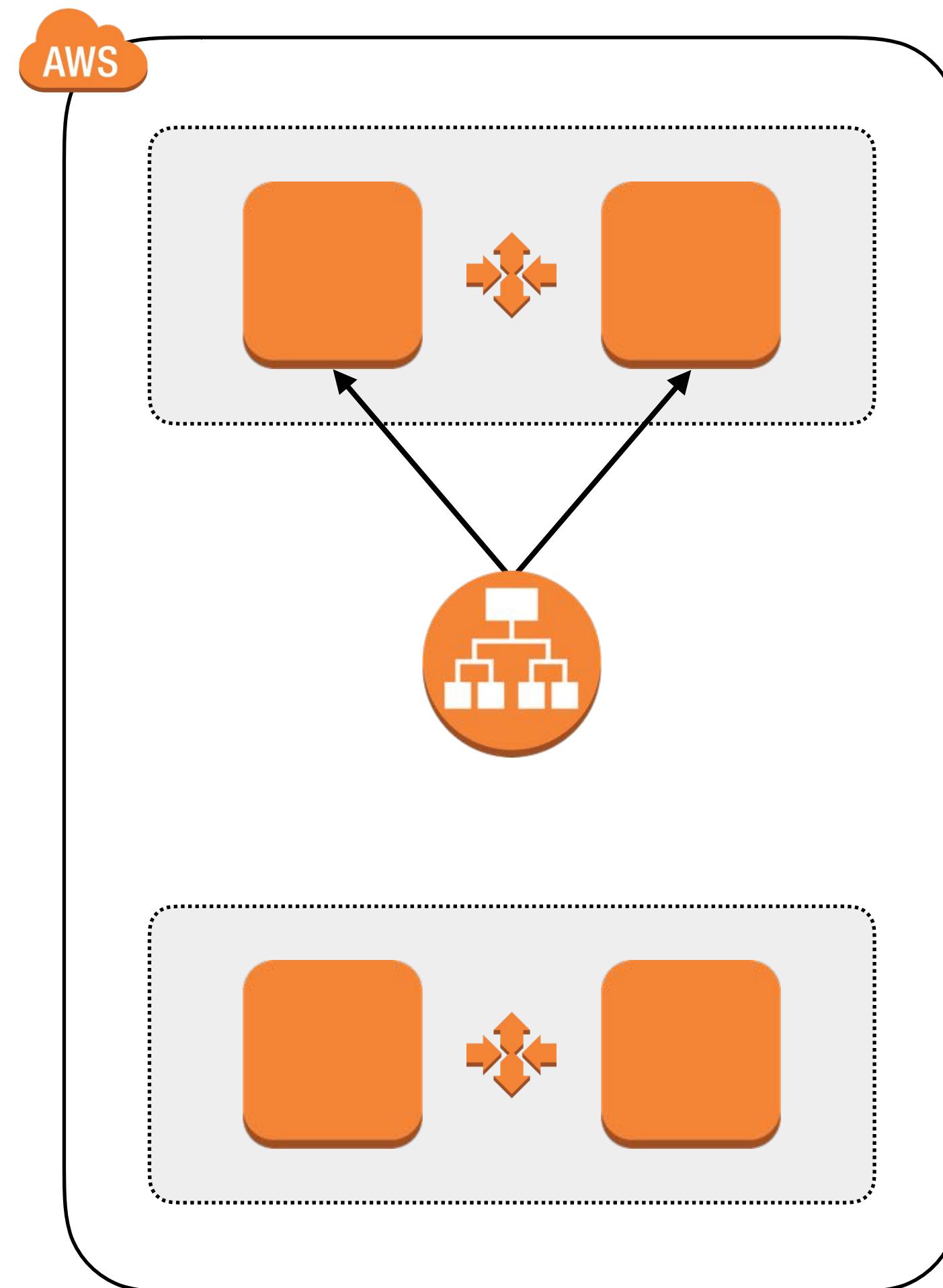
- In-place deployment
- **Blue/green deployment**



CodeDeploy - Deployment type



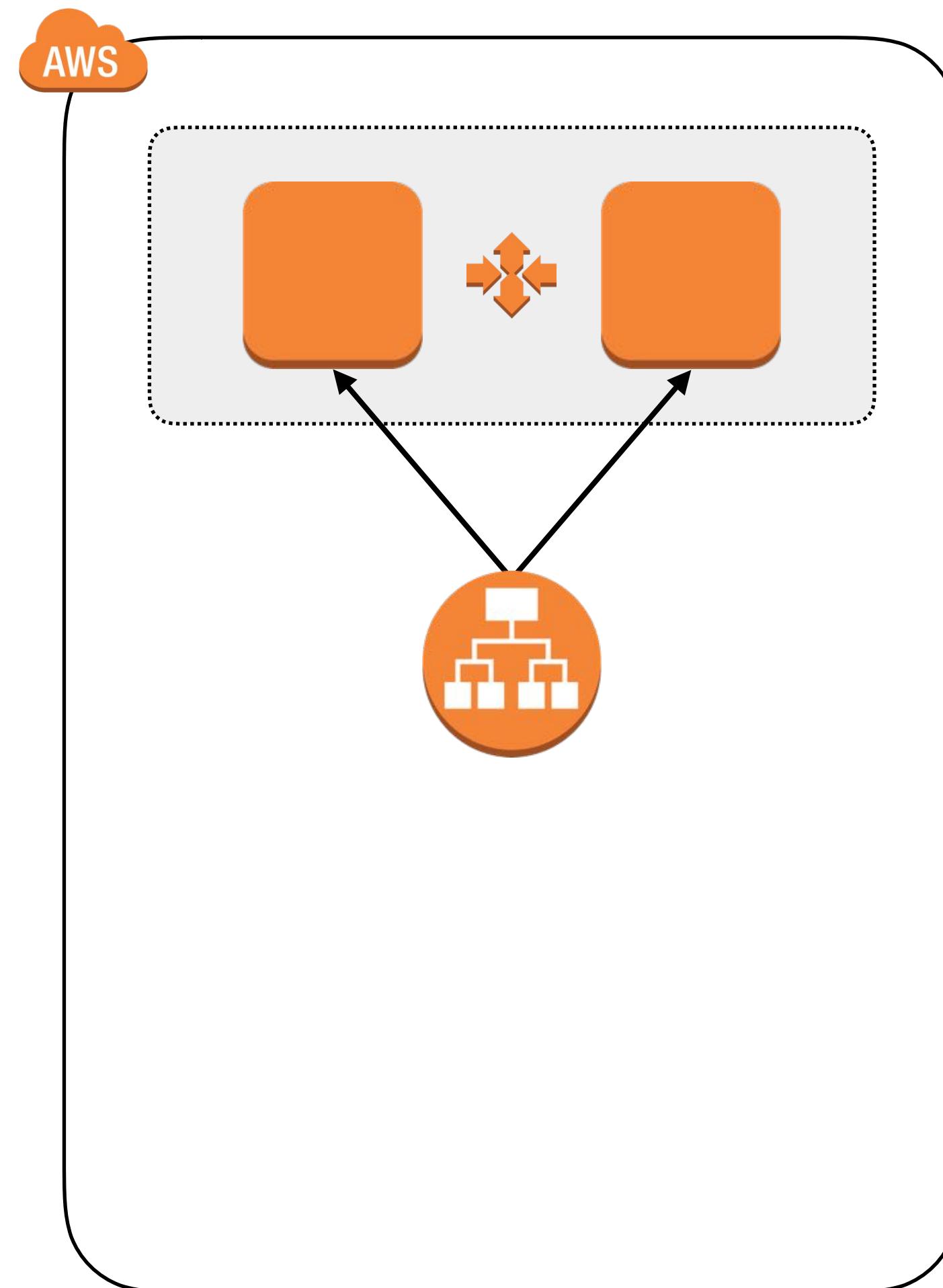
- In-place deployment
- **Blue/green deployment**



CodeDeploy - Deployment type



- In-place deployment
- **Blue/green deployment**





CodeDeploy - Deployment type



In-place deployment

Updates the instances in the deployment group with the latest application revision. During a deployment, each instance will be briefly taken offline for its update.



Blue/green deployment

Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

CodeDeploy - Environment configuration



Automatically copy Auto Scaling group

Provision an Auto Scaling group and deploy the new application revision to it. AWS CodeDeploy will create the Auto Scaling group by copying the one you specify here.



Manually provision instances

I will specify here the instances where the current application revision is running. I will specify the instances for the replacement environment when I create a deployment.

CodeDeploy - Environment configuration



Auto Scaling group* Choose the Auto Scaling group where the current application revision is deployed.

A screenshot of the AWS CodeDeploy environment configuration interface. A dropdown menu titled "Choose an Auto Scaling group" is open, showing a list of available groups. The group "demo-application" is highlighted in orange at the bottom of the list, indicating it is selected. Above the dropdown, there is a search bar labeled "Filter by Auto Scaling group name".

CodeDeploy - Environment configuration



Enable load balancing

Load balancer* Select the load balancer that is currently routing traffic to your original environment. After a blue/green deployment is complete, the load balancer you select will route all traffic to the replacement environment.



Application Load Balancer or Network Load Balancer

demo-application



CodeDeploy - Deployment settings



Deployment settings

The blue/green deployment will replace the instances in the deployment group using the following settings:

Traffic rerouting	Reroute traffic immediately
Deployment configuration	Deploy to one instance at a time
Original instances	Terminate the original instances in the deployment group after 0 days 1 hours 0 minutes

Want to use different settings? [Edit deployment settings](#)

CodeDeploy - Deployment settings



Deployment settings

The blue/green deployment will replace the instances in the deployment group using the following settings:

Deployment configuration Deploy to one instance at a time

Deployment configuration: Terminates the original instances in the deployment group after 0 days 0 hours 0 minutes

Want to use different settings? [Edit deployment settings](#)

CodeDeploy - Deployment settings



Deployment settings

The blue/green deployment will replace the instances in the deployment group using the following settings:

Traffic rerouting

Reroute traffic immediately

minutes

Want to use different settings? [Edit deployment settings](#)

CodeDeploy - Deployment settings



Deployment settings

The blue/green deployment will replace the instances in the deployment group using the following settings:

Traffic routing: [Route traffic immediately](#)

Original instances	Terminate the original instances in the deployment group after 0 days 1 hours 0 minutes
---------------------------	-----------------------------------------------------------------------------------------

minutes

Want to use different settings? [Edit deployment settings](#)



CodeDeploy - Deployment settings

Deployment settings

The blue/green deployment will replace the instances in the deployment group using the following settings:

Traffic rerouting Reroute traffic immediately

Deployment configuration Deploy to one instance at a time

Original instances Terminate the original instances in the deployment group after 0 days 1 hours 0 minutes

Want to use different settings? [Edit deployment settings](#)

CodeDeploy - Edit deployment settings



Service role ARN*

arn:aws:iam::012345678910:role/CodeDeployServiceRole  ▾



appspec.yml

CodeDeploy - appspec.yml



```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/demo-application/
permissions:
  - object: /var/www/demo-application
    owner: root
    group: nginx
    mode: 750
    type:
      - directory
  - object: /var/www/demo-application
    owner: root
    group: nginx
    mode: 640
    type:
      - file
hooks:
  BeforeInstall:
    - location: .aws/codedeploy/hooks/before-install.sh
      timeout: 300
      runas: root
```



CodeDeploy - appspec.yml

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/demo-application/
permissions:
  - object: /var/www/demo-application
    owner: root
    group: nginx
    mode: 644
    type:
      - file
  - object: /var/www/demo-application/.htaccess
    owner: root
    group: nginx
    mode: 640
    type:
      - file
hooks:
  BeforeInstall:
    - location: .aws/codedeploy/hooks/before-install.sh
      timeout: 300
      runas: root
```

version: 0.0



CodeDeploy - appspec.yml

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/demo-application/
permissions:
  - object: /var/www/demo-application
    owner: root
    group: nginx
    mode: 755
    type:
      - directory
  - object:
    owner: root
    group: nginx
    mode: 640
    type:
      - file
hooks:
  BeforeInstall:
    - location: .aws/codedeploy/hooks/before-install.sh
      timeout: 300
      runas: root
```

os : linux



CodeDeploy - appspec.yml

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/demo-application/
permissions:
  object: /var/www/demo_application
```

```
files:
  - source: /
    destination: /var/www/demo-application/
```

```
type:
  - file
hooks:
https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-files.html
```



CodeDeploy - appspec.yml

```
version: 0.0
os: Ubuntu-14.04
files:
  - file:
    permissions:
      - object: /var/www/demo-application
        owner: root
        group: nginx
        mode: 750
        type:
          - directory
      - object: /var/www/demo-application
        owner: root
        group: nginx
        mode: 640
        type:
          - file
  - file
hooks:
```

[https://docs.aws.amazon.com/codedeploy/latest/userguide/
reference-appspec-file-structure-permissions.html](https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-permissions.html)



CodeDeploy - appspec.yml

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/demo-application/
permissions:
  - object: /var/www/demo-application

hooks:
  BeforeInstall:
    - location: .aws/codedeploy/hooks/before-install.sh
      timeout: 300
      runas: root

      mode: 0700
      type: file

#!/bin/bash

rm -rf /var/www/demo-application/*
```



CodeDeploy - appspec.yml

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/demo-application/
permissions:
  - object: /var/www/demo-application
```

hooks:

BeforeInstall:

- location: .aws/codedeploy/hooks/before-install.sh
timeout: 300
runas: root

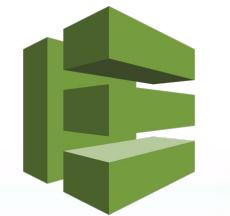
mode: 0700

type:

- file

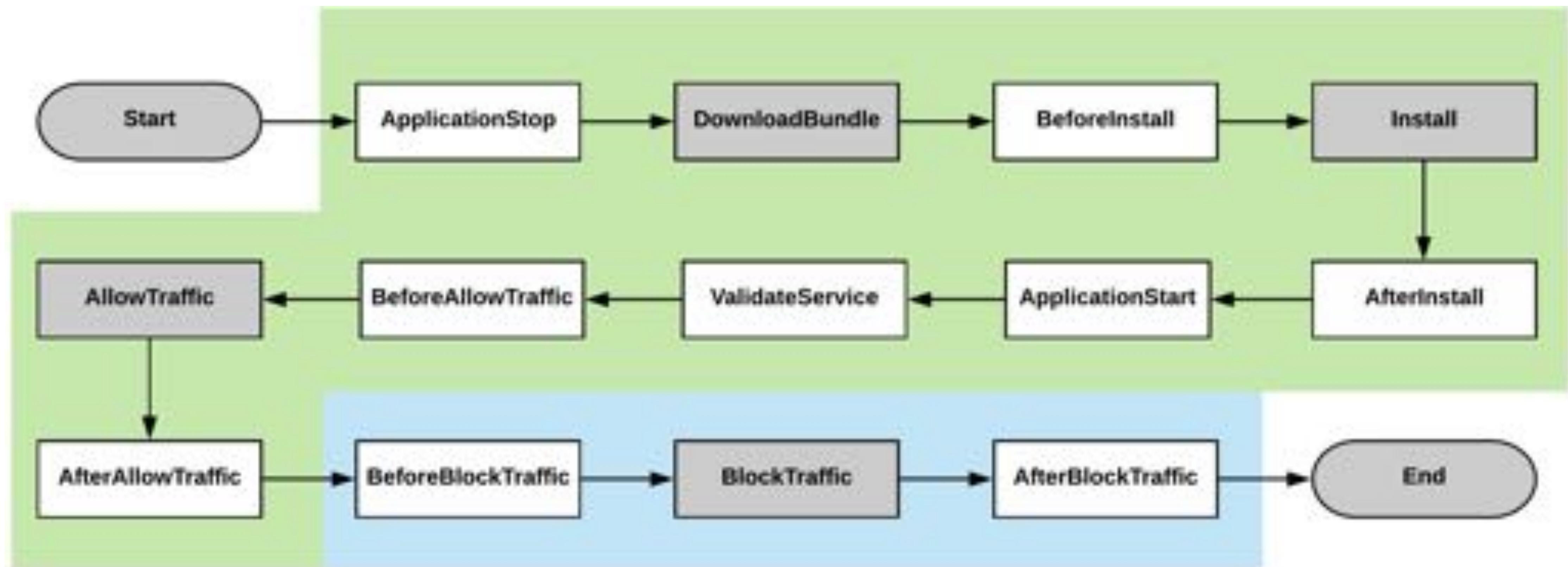
<https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html>

CodeDeploy - appspec.yml



```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/demo-application/
permissions:
  - object: /var/www/demo-application
    owner: root
    group: nginx
    mode: 750
    type:
      - directory
  - object: /var/www/demo-application
    owner: root
    group: nginx
    mode: 640
    type:
      - file
hooks:
  BeforeInstall:
    - location: .aws/codedeploy/hooks/before-install.sh
      timeout: 300
      runas: root
```

CodeDeploy - appspec.yml - Event lifecycle



CodeDeploy - Deploy a revision



Application*

DemoApplication



Compute platform

EC2/On-premises

Deployment group*

DemoApplication



Deployment type

Blue/green



Repository type*



My application is stored in Amazon S3



My application is stored in GitHub

CodeDeploy - Deploy a revision



Deployment description

Our first deployment!

Deployment configuration

CodeDeployDefault.AllAtOnce



CodeDeploy - Deploy a revision



GitHub account*

michaelmoussa

[Connect to a different GitHub account](#)

Repository Name*

<organization-name>/<repository-name>

Commit ID*

<full 40 character git commit hash>

CodeDeploy - Deploy a revision



Fail the deployment

An error is reported and the deployment status is changed to Failed.



Overwrite the content

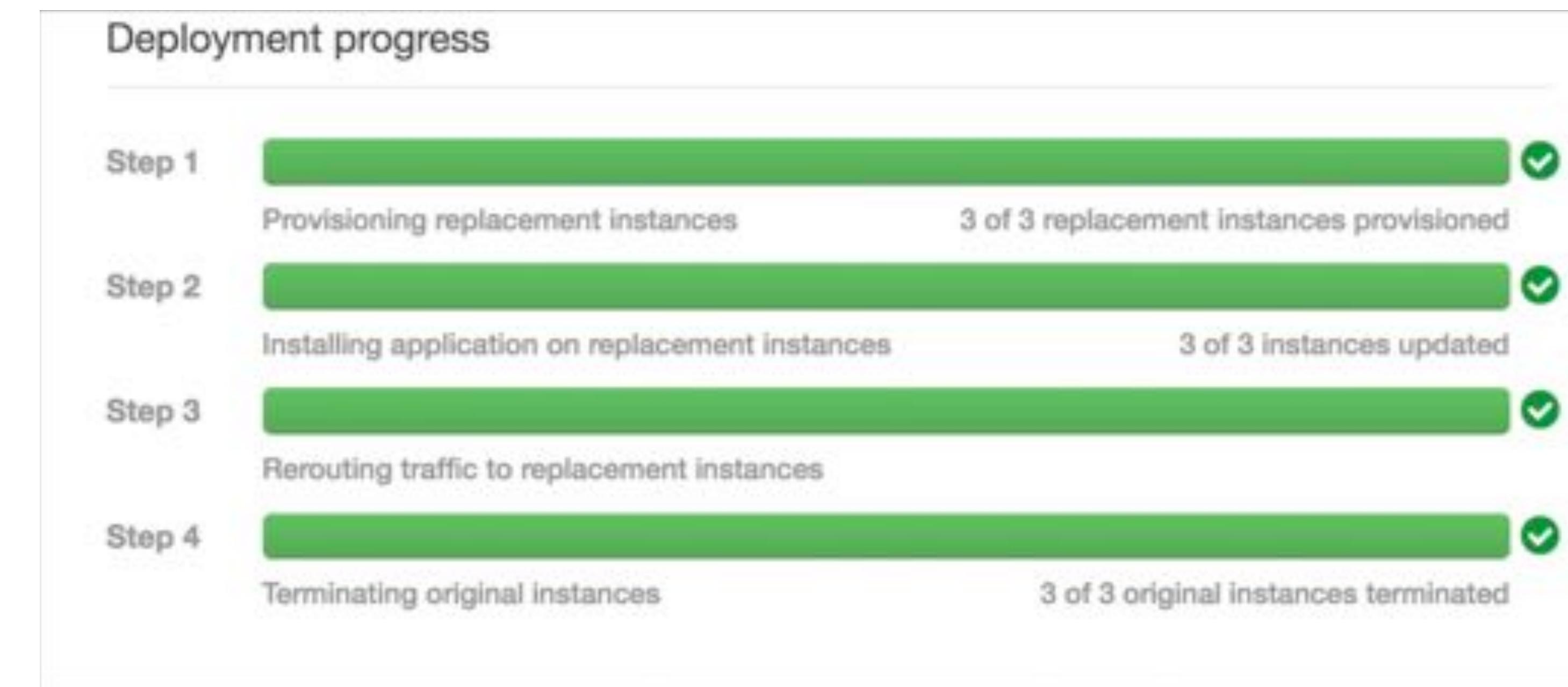
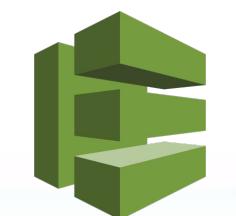
The file in the application revision is copied to the target location on the instance, replacing the previous file.



Retain the content

The file in the application revision is not copied to the instance. The existing file is kept at the target location and treated as part of the new deployment.

CodeDeploy - Deploy a revision





Other capabilities



CodeDeploy - Other capabilities



- **SNS Triggers**
 - Deployment (start | succeeds | fails | stops | ready | rollback)
 - Instance (starts | succeeds | fails | ready)

CodeDeploy - Other capabilities



- **SNS Triggers**
 - Deployment (start | succeeds | fails | stops | ready | rollback)
 - Instance (starts | succeeds | fails | ready)
- **CloudWatch Alarms**
 - Stop deployments if metric threshold is breached

CodeDeploy - Other capabilities

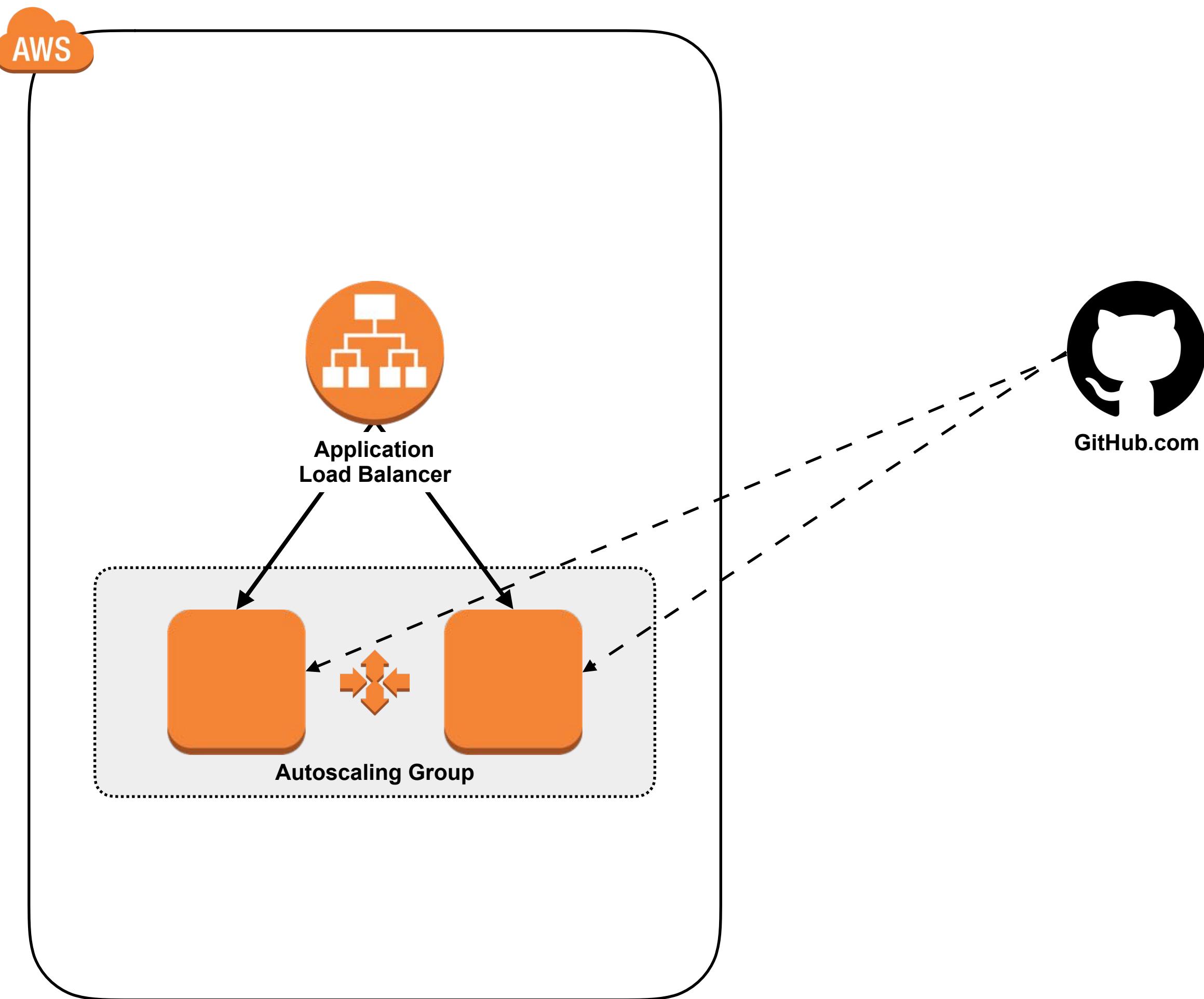


- **SNS Triggers**
 - Deployment (start | succeeds | fails | stops | ready | rollback)
 - Instance (starts | succeeds | fails | ready)
- **CloudWatch Alarms**
 - Stop deployments if metric threshold is breached
- **Automatic Rollback**

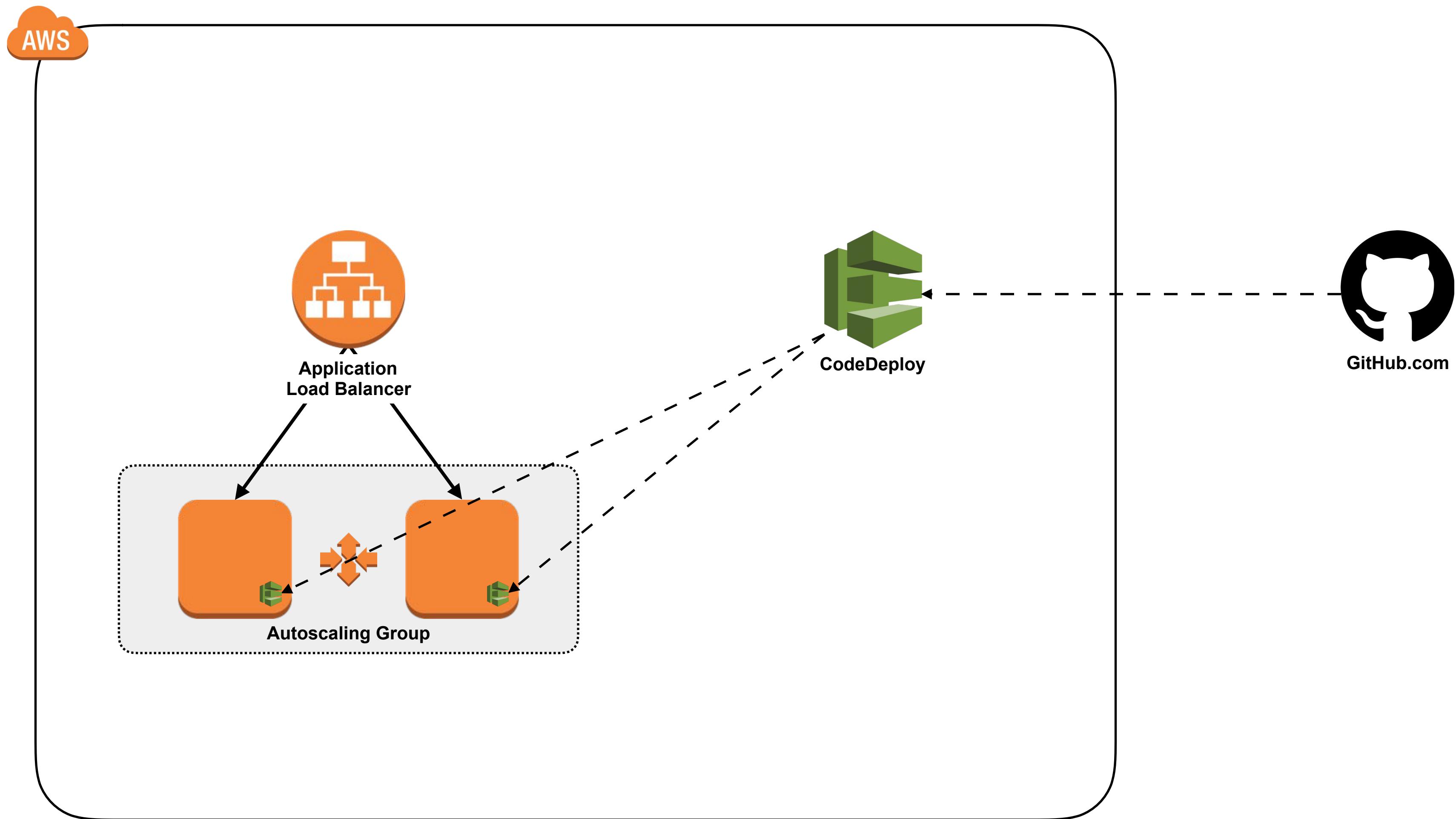
Objective

Set up CICD for a non-trivial PHP application
running on AWS, and deploy it with zero downtime.

Objective



Objective



Objective

Set up CI/CD for a *non-trivial* PHP application
running on AWS, and deploy it with zero downtime.

CodeBuild



CodeBuild

- Fully managed build service



CodeBuild

- Fully managed build service
 - Compiles, tests, and packages code



CodeBuild

- Fully managed build service
 - Compiles, tests, and packages code
 - Scales continuously



CodeBuild

- Fully managed build service
 - Compiles, tests, and packages code
- Scales continuously
- Pay-as-you-go



CodeBuild

- Fully managed build service
 - Compiles, tests, and packages code
- Scales continuously
- Pay-as-you-go
- Also highly customizable!



CodeBuild - Get started



Project name*

DemoApplication



Description

Remove description

An AWS CodeDeploy PHP demo application!

255 characters max

CodeBuild - Source



Source provider*

Choose a source provider

|

Amazon S3

AWS CodeCommit

Bitbucket

GitHub

GitHub Enterprise

CodeBuild - Source



Source provider*

GitHub

Repository*

- Use a public repository
- Use a repository in my account

Choose a repository

<organization-name>/<repository-name>



[Disconnect from GitHub](#)

CodeBuild - Environment



Environment image*



Use an image managed by
AWS CodeBuild



Specify a Docker image

CodeBuild - Environment



Operating system*

Ubuntu

Runtime*

PHP

CodeBuild - Environment



Runtime version*

Choose a runtime environ...

aws/codebuild/php:5.6

aws/codebuild/php:7.0

CodeBuild - Environment



Environment image* Use an image managed by AWS CodeBuild
 Specify a Docker image

Environment type* Linux

Custom image type* Other

Custom image ID* amazonlinux:2017.12.0.20180330 i

CodeBuild - Environment



Environment image* Use an image managed by AWS
CodeBuild

Custom image type*

Other

Custom image ID*

amazonlinux:2017.12.0.20180330



Custom image ID*

amazonlinux:2017.12.0.20180330



CodeBuild - buildspec.yml



Build specification



Use the buildspec.yml in the source code root directory



Insert build commands

Buildspec name

buildspec.yml



CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y
      - yum install git-core php-mbstring php-xml zip unzip php-zip -y
      - curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php
      - php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
  pre_build:
    commands:
      - composer install --no-interaction
  build:
    commands:
      - composer cs-check
      - composer test
  post_build:
    commands:
      - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y
      - yum install git-core php-mbstring php-xml zip unzip php-zip -y
      - curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php
      - php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
  pre_build:
    commands:
      - composer install --
  build:
    commands:
      - composer cs-check
      - composer test
  post_build:
    commands:
      - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

version: 0.2



CodeBuild - buildspec.yml

```
version: 0.2

phases:
  install:
    commands:

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y
      - yum install git-core php-mbstring php-xml zip unzip php-zip -y
      - curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php
      - php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
    post_build:
      commands:
        - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y
      - yum install git-core php-mbstring php-xml zip unzip php-zip -y
      - curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php
      - php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
    post_build:
      commands:
        - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y
      - yum install git-core php-mbstring php-xml zip unzip php-zip -y
      - curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php
      - php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
      post_build:
        commands:
          - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:
      - yum update -y

phases:

  pre_build:
    commands:
      - composer install --no-interaction

  post_build:
    commands:
      - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction

artifacts:
  files:
    - '**/*'
```

CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:
      - yum upd
      - amazon-
      - yum ins
      - curl -s
      - php /tm
  pre_build:
    commands:
      - composer
  build:
    commands:
      - composer
      - composer
  post_build:
    commands:
      - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

phases:

build:

commands:

- composer cs-check
- composer test

CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y

phases:

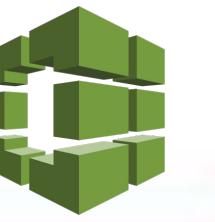
post_build:
  commands:
    - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
      - composer cs-check
      - composer test
    post_build:
      commands:
        - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y
      - yum install git-core php-mbstring php-xml unzip php-zip -y
      - curl -sS https://getcomposer.org/installer | php -- --filename=composer
      - php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
  pre_build:
    commands:
      - composer install --no-dev
  build:
    commands:
      - composer cs-check
      - composer test
  post_build:
    commands:
      - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```



CodeBuild - buildspec.yml

```
version: 0.2

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y
      - yum install git-core php-mbstring php-xml zip unzip php-zip -y
      - curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer
  pre_build:
    commands:
      - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
  build:
    commands:
      - composer cs-check
      - composer test
  post_build:
    commands:
      - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

<https://docs.aws.amazon.com/codebuild/latest/userguide/build-spec-ref.html>

CodeBuild - buildspec.yml



```
version: 0.2

phases:
  install:
    commands:
      - yum update -y
      - amazon-linux-extras install php7.2 -y
      - yum install git-core php-mbstring php-xml zip unzip php-zip -y
      - curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php
      - php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
  pre_build:
    commands:
      - composer install --no-interaction
  build:
    commands:
      - composer cs-check
      - composer test
  post_build:
    commands:
      - composer install --no-dev --optimize-autoloader --prefer-dist --no-interaction
artifacts:
  files:
    - '**/*'
```

CodeBuild - Artifacts



Type*	Amazon S3	
Name	DemoApplication.zip	
Path	build-artifacts	
Namespace type	Build ID	
Bucket name*	php-codedeploy-example	

CodeBuild - Artifacts



Type* Amazon S3 i

Type* Amazon S3 i

Path build-artifacts i

Bucket name* php-codedeploy-exa... refresh

Bucket name* php-codedeploy-example refresh

CodeBuild - Artifacts



Type* Amazon S3 i

Name	DemoApplication.zip i
Path	build-artifacts i

Bucket name* php-codedeploy-example ↻

CodeBuild - Artifacts



Type* Amazon S3 i

Name DemoApplication.zip i

Namespace type Build ID i

Namespace type Build ID i

Bucket name* php-codedeploy-example ↻

CodeBuild - Artifacts



Type*	Amazon S3	
Name	DemoApplication.zip	
Path	build-artifacts	
Namespace type	Build ID	
Bucket name*	php-codedeploy-example	

CodeBuild - Artifacts

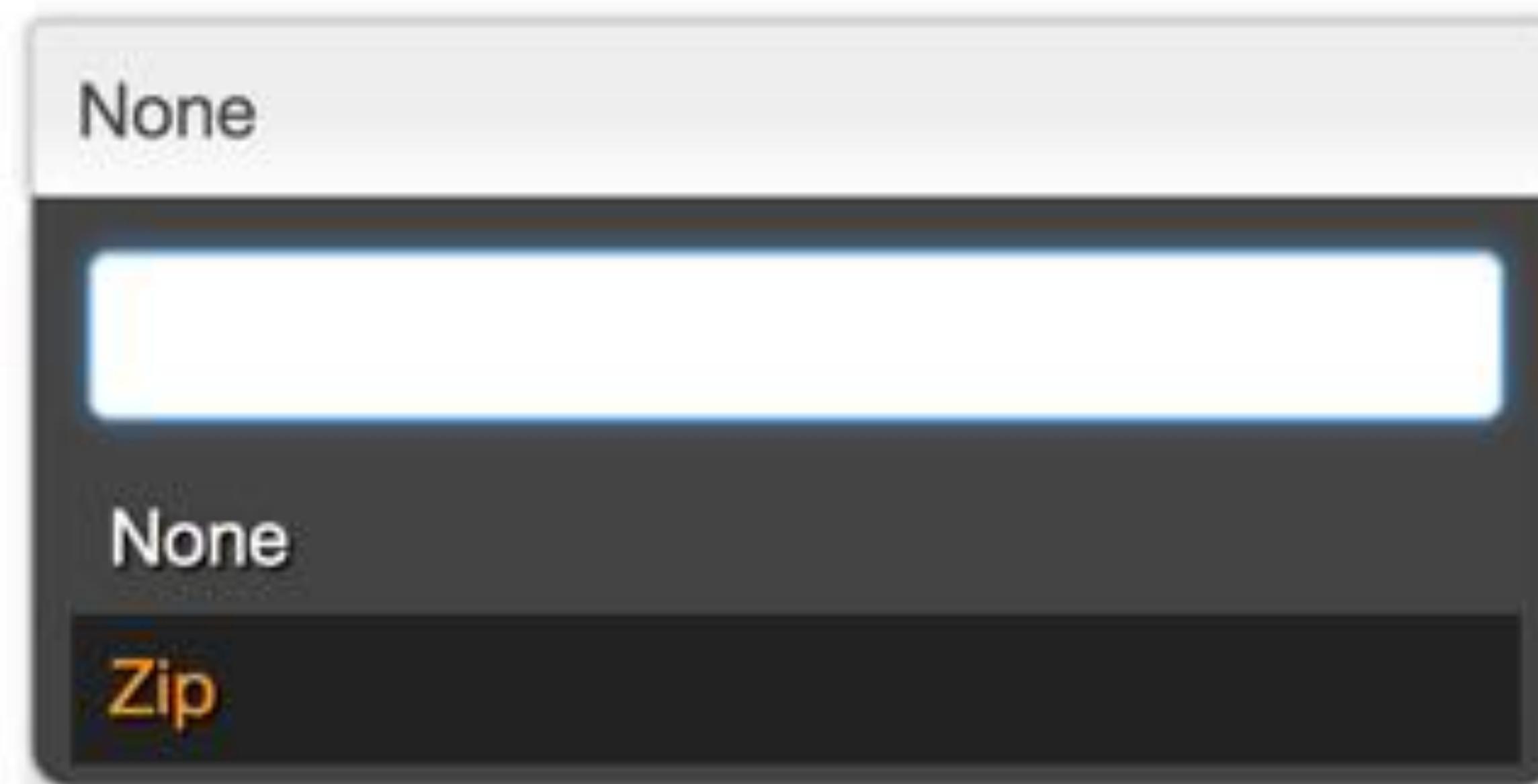


s3://php-codedeploy-example
/build-artifacts
/c4985014-e610-43ed-a5d6-551c47570c8d
/DemoApplication.zip

CodeBuild - Artifacts packaging



Artifacts packaging



CodeBuild - Start build



Start build

Create project

Actions ▾

Project	Repository
 DemoApplication	https://github.com/mich

CodeBuild - Start build



Source version

master



CodeBuild - Success!



Name	Status	Duration	Completed
SUBMITTED	Succeeded		3 minutes ago
PROVISIONING	Succeeded	31 secs	2 minutes ago
DOWNLOAD_SOURCE	Succeeded	58 secs	60 seconds ago
INSTALL	Succeeded	39 secs	21 seconds ago
PRE_BUILD	Succeeded	7 secs	13 seconds ago
BUILD	Succeeded		12 seconds ago
POST_BUILD	Succeeded		12 seconds ago
UPLOAD_ARTIFACTS	Succeeded		11 seconds ago
FINALIZING	Succeeded	2 secs	9 seconds ago
COMPLETED	Succeeded		



Other capabilities

CodeBuild - Other capabilities



- Dependency caching

CodeBuild - Other capabilities



- Dependency caching
- VPC support

CodeBuild - Other capabilities



- Dependency caching
- VPC support
- Environment variables & secure parameters

CodeBuild - Other capabilities



- Dependency caching
- VPC support
- Environment variables & secure parameters
- Build Docker images

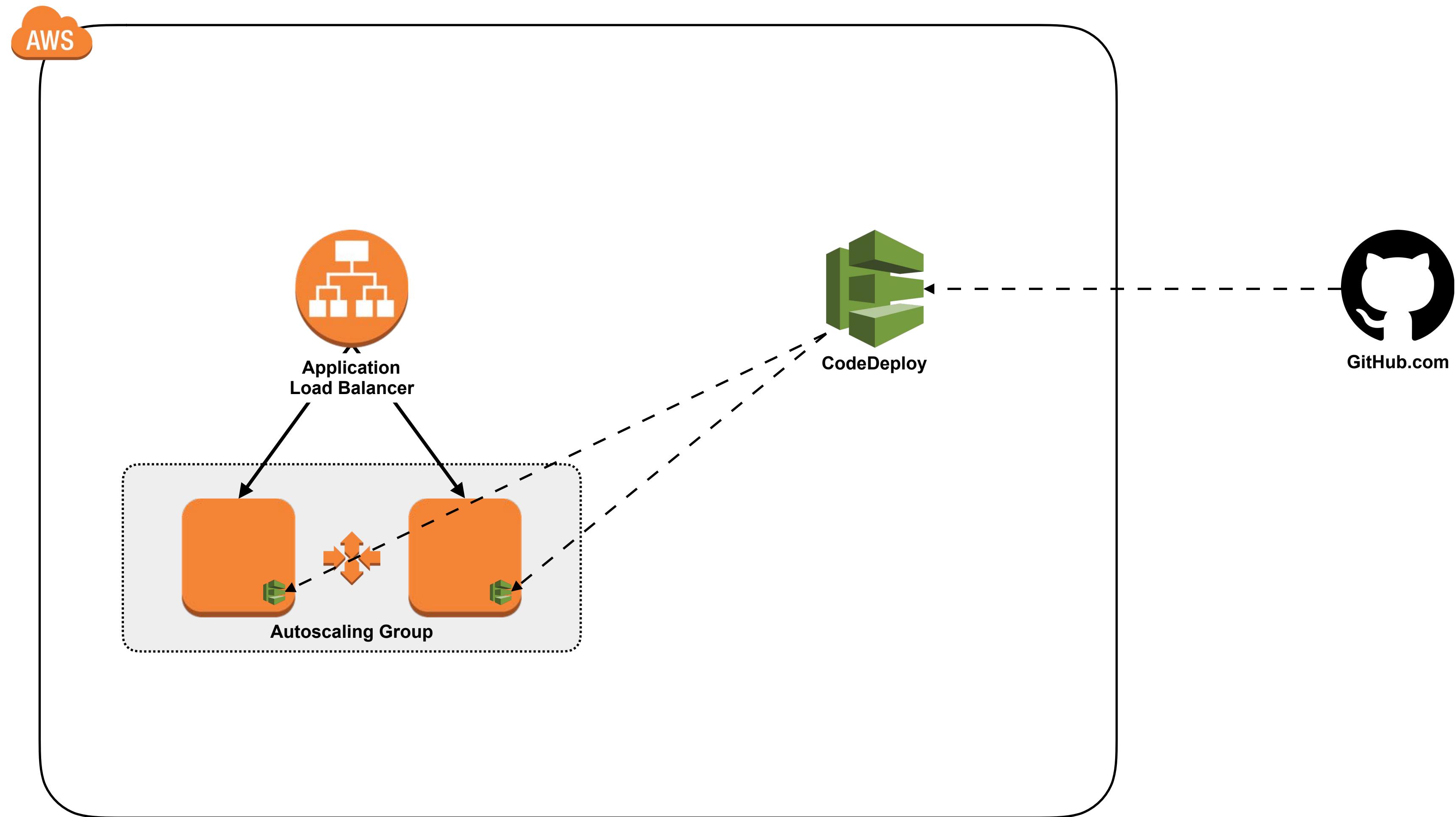
CodeBuild - Other capabilities



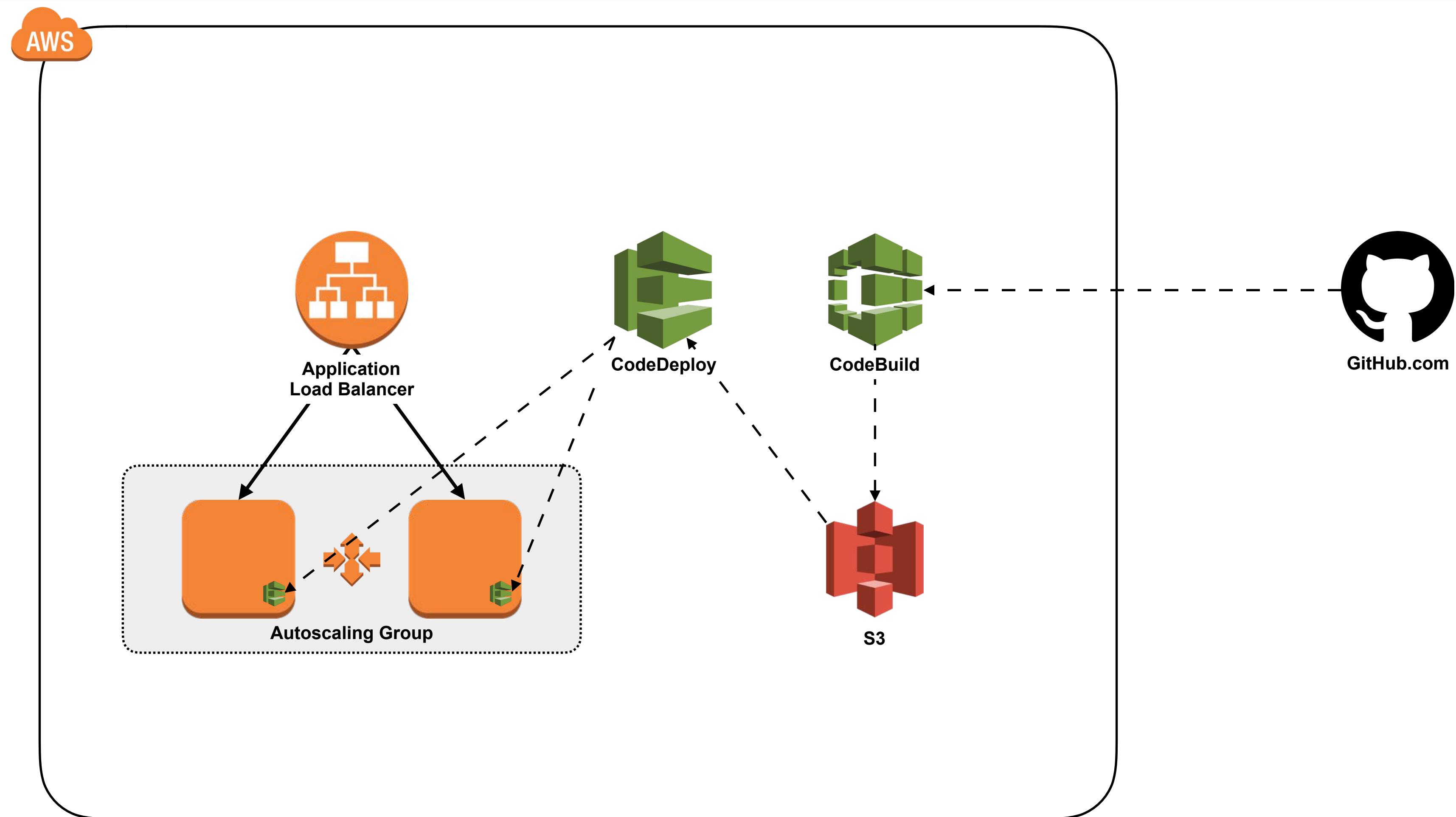
- Dependency caching
- VPC support
- Environment variables & secure parameters
- Build Docker images
- Larger build instance sizes

CodeBuild → CodeDeploy

CodeBuild → CodeDeploy



CodeBuild → CodeDeploy



CodeBuild → CodeDeploy

Repository type*



My application is stored in Amazon S3



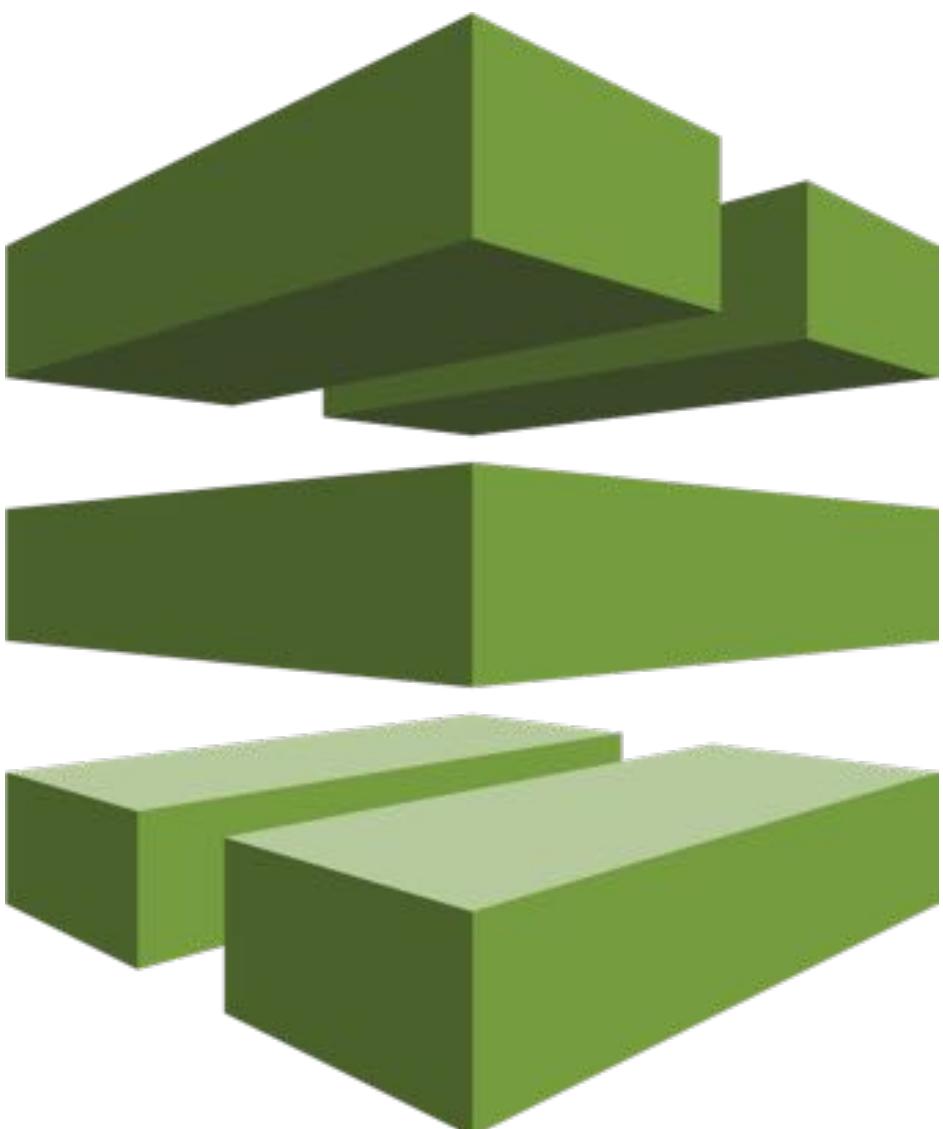
My application is stored in GitHub

Revision location*

s3://php-codedeploy-example/build-artifacts/c498f▼

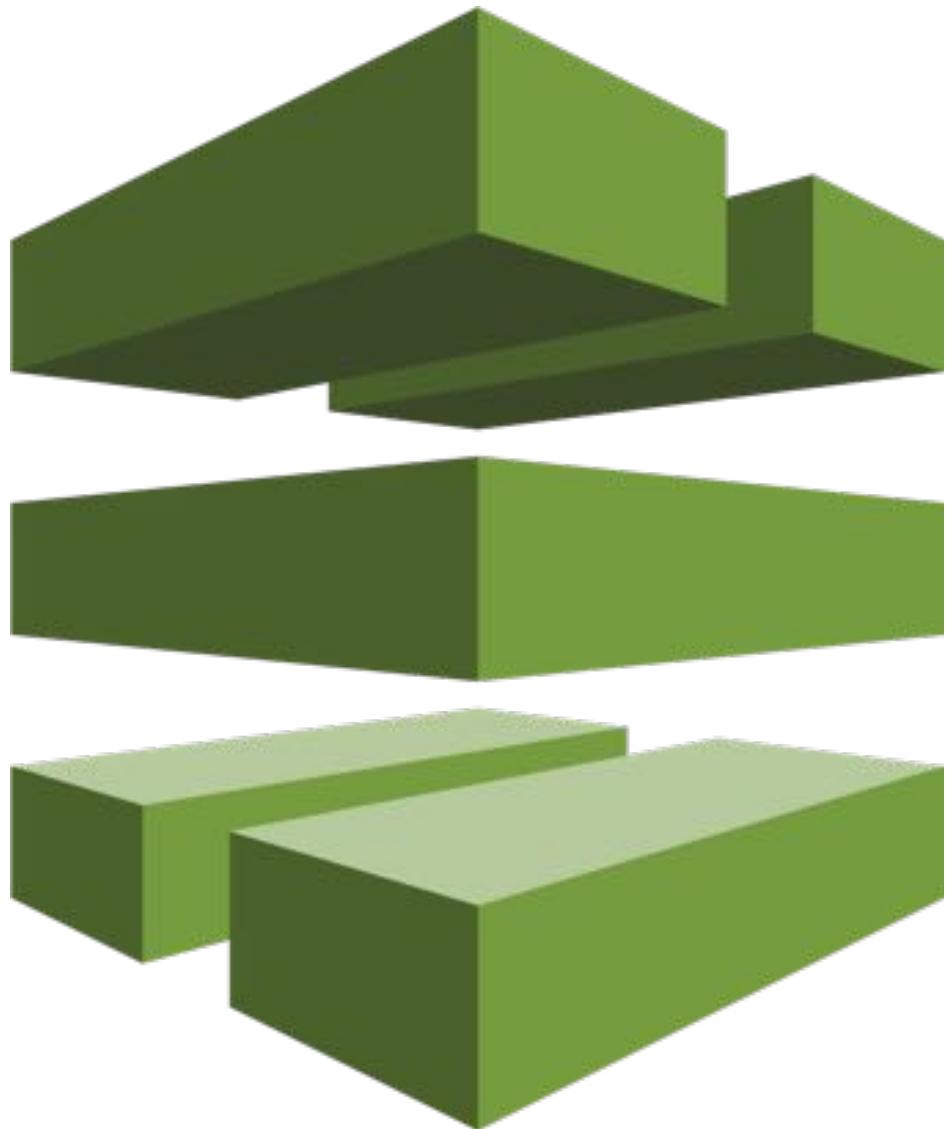


CodePipeline



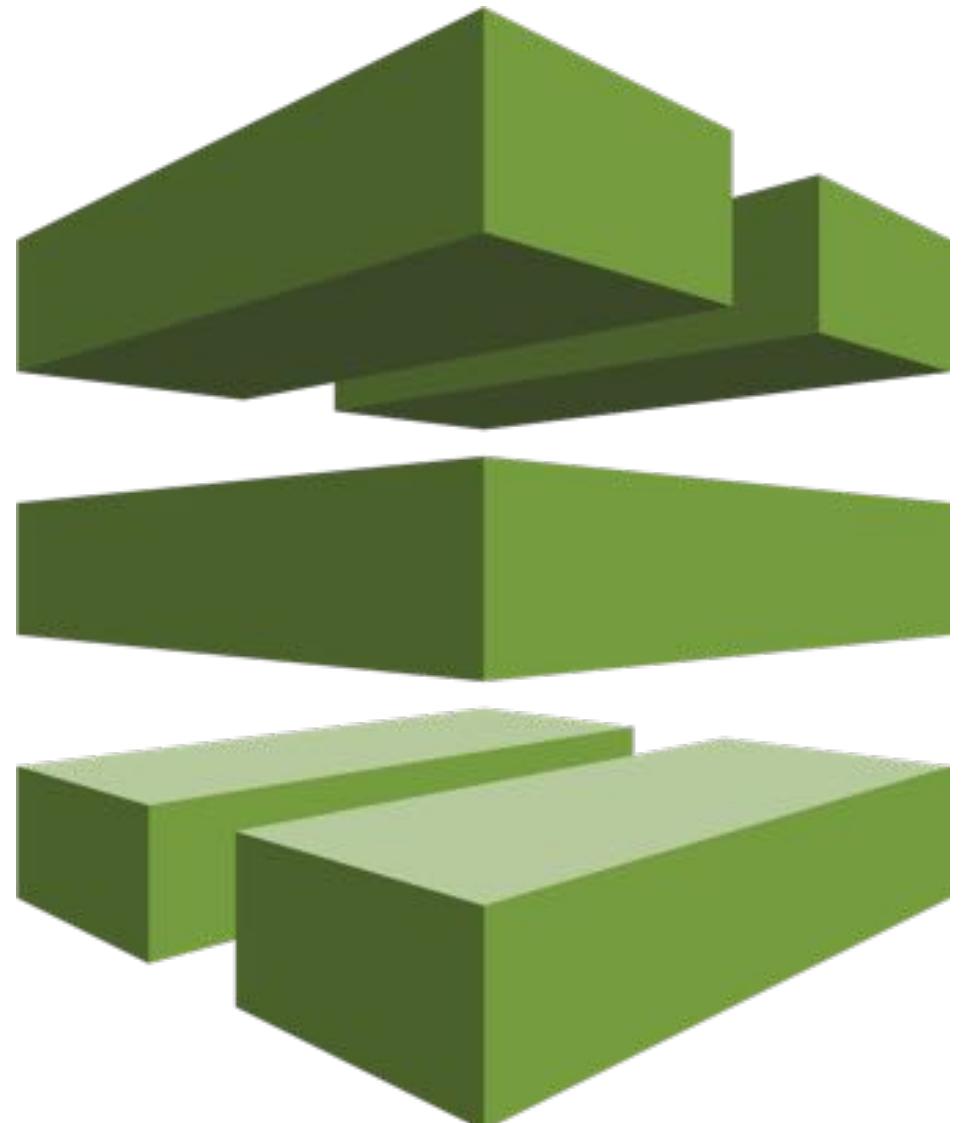
CodePipeline

- Continuous integration & continuous delivery service



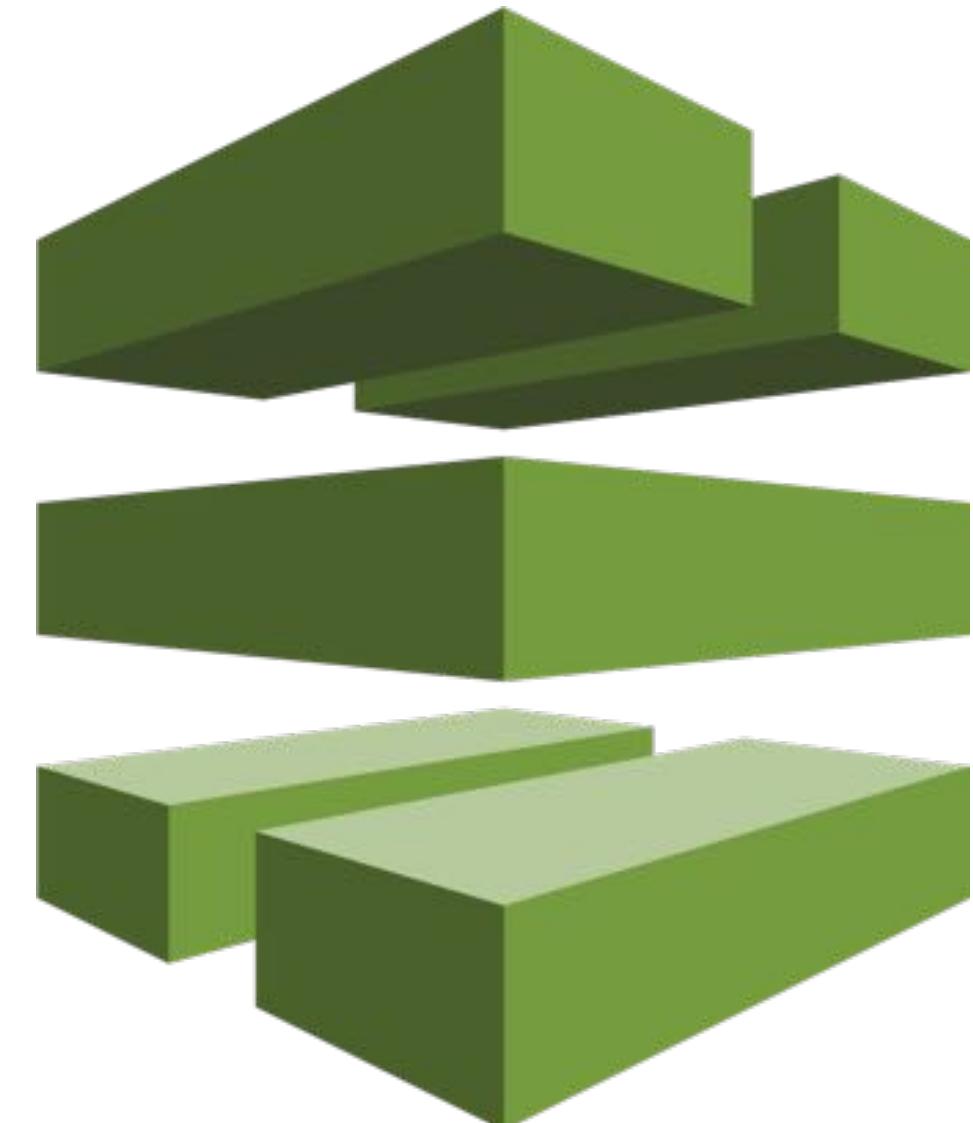
CodePipeline

- Continuous integration & continuous delivery service
 - Model, visualize, and automate steps



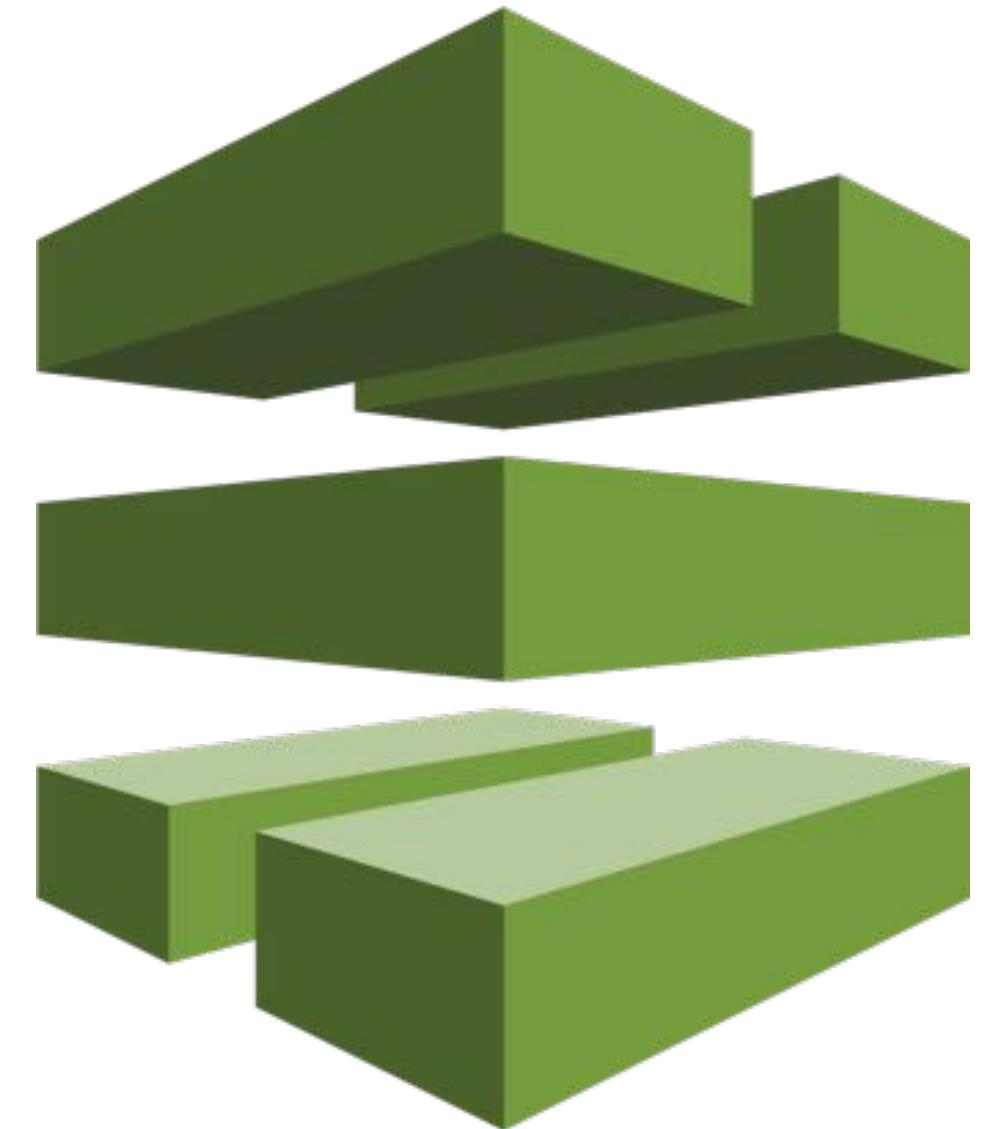
CodePipeline

- Continuous integration & continuous delivery service
 - Model, visualize, and automate steps
 - AWS & third-party integration support



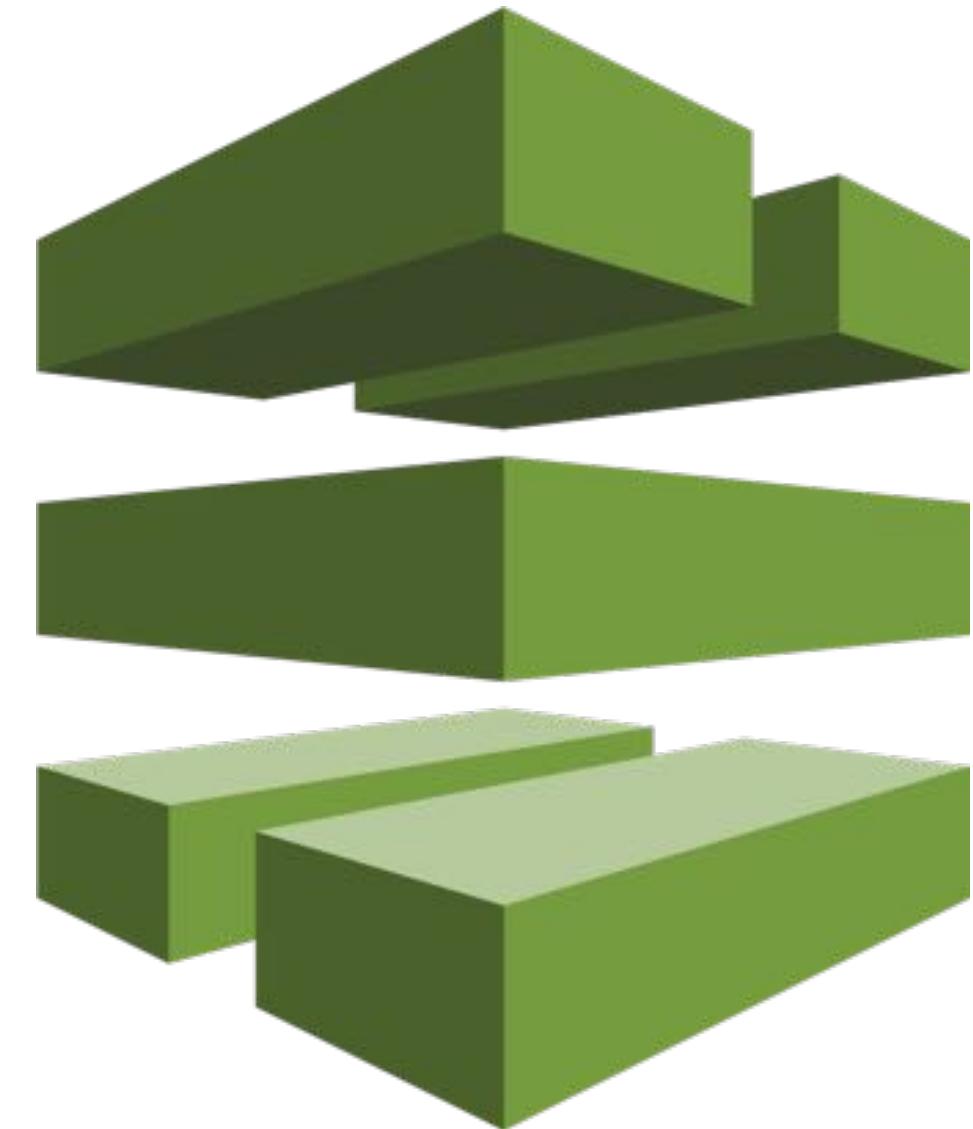
CodePipeline

- Continuous integration & continuous delivery service
 - Model, visualize, and automate steps
- AWS & third-party integration support
- Fully configurable workflow

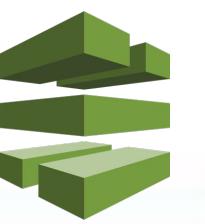


CodePipeline

- Continuous integration & continuous delivery service
 - Model, visualize, and automate steps
- AWS & third-party integration support
- Fully configurable workflow
- Cheap!



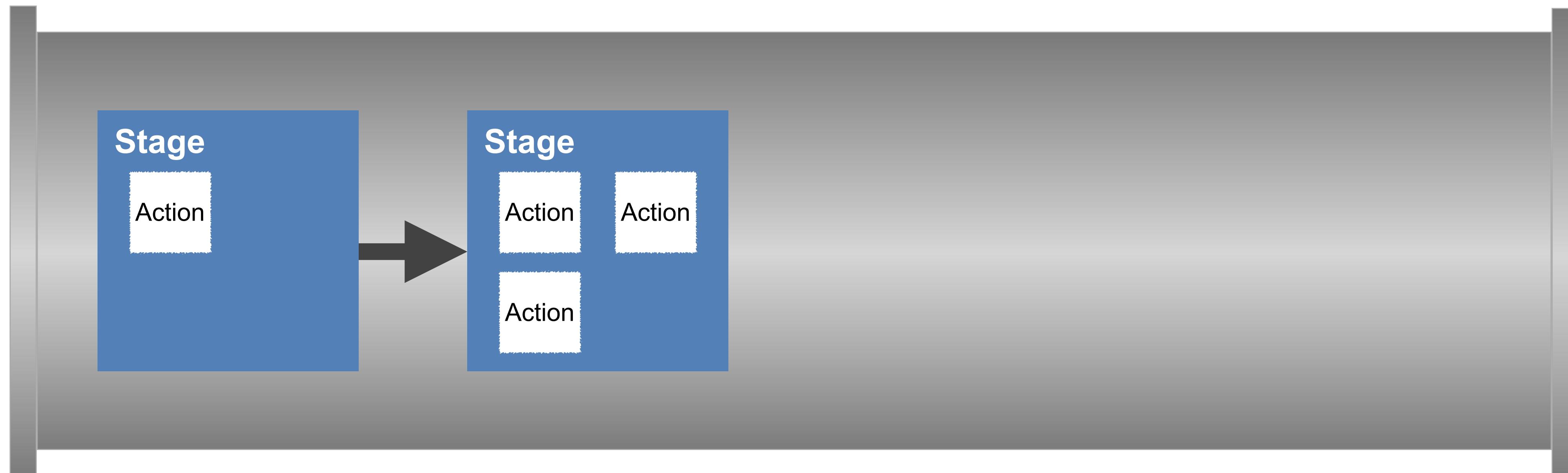
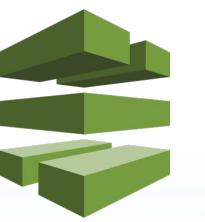
CodePipeline



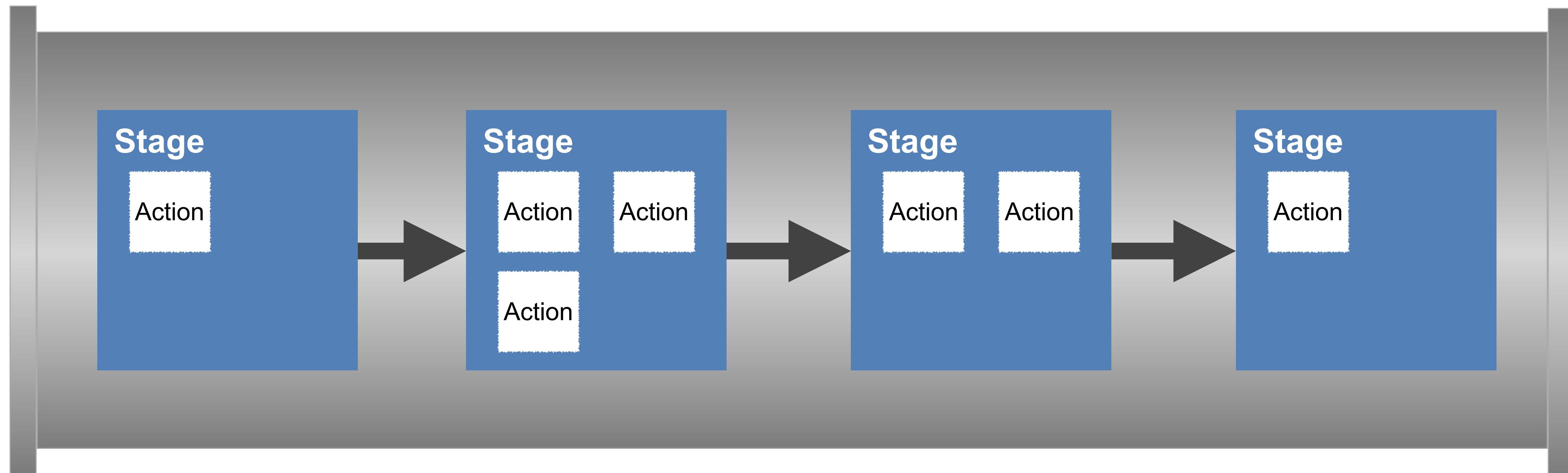
CodePipeline



CodePipeline



CodePipeline



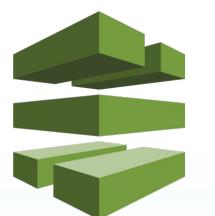
CodePipeline - Create pipeline



Pipeline name*

DemoApplication

CodePipeline - Source location



Repository*

<organization-name>/<repository-name>

Branch*

master

CodePipeline - Build provider



Select an existing build project



Create a new build project

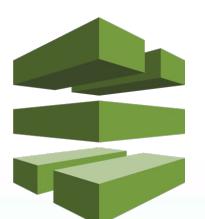
Project name*

DemoApplication



View project details

CodePipeline - Deployment provider



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

Application name*

DemoApplication



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

Deployment group*

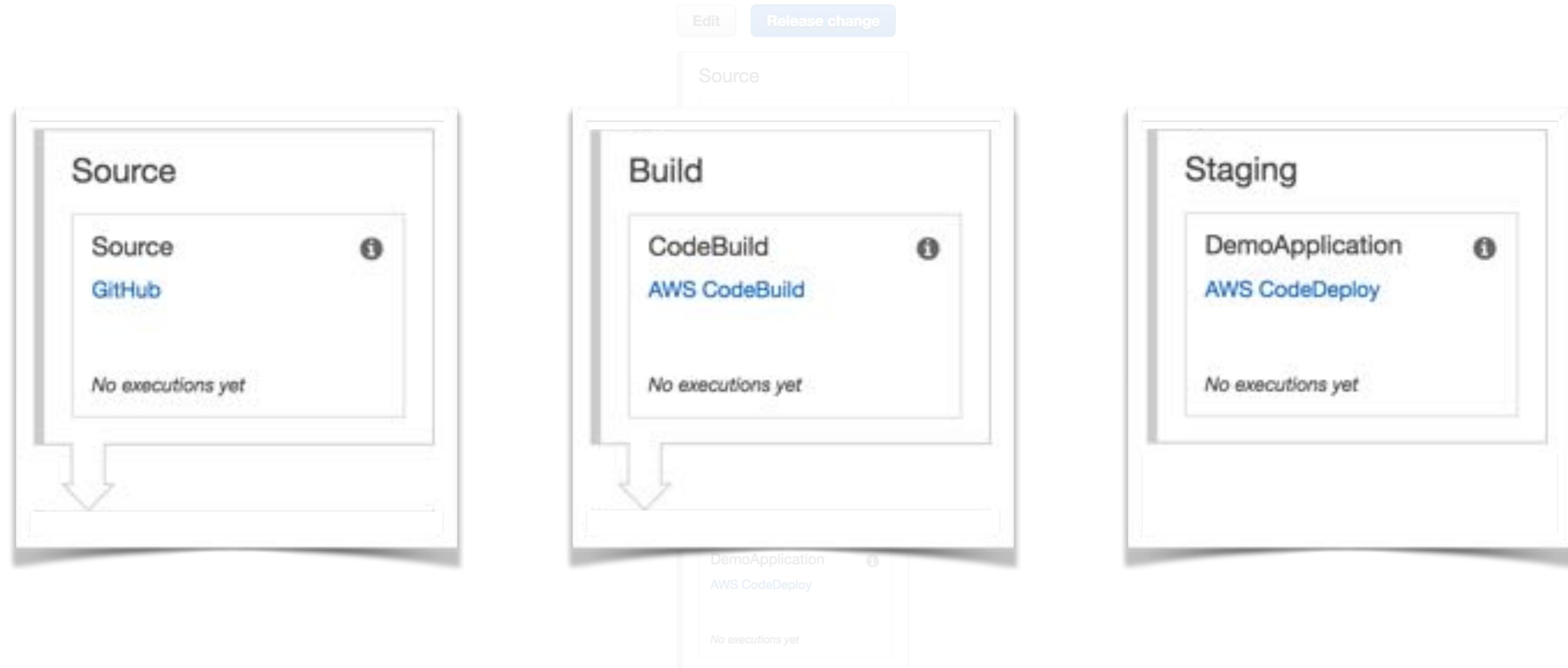
DemoApplication



CodePipeline

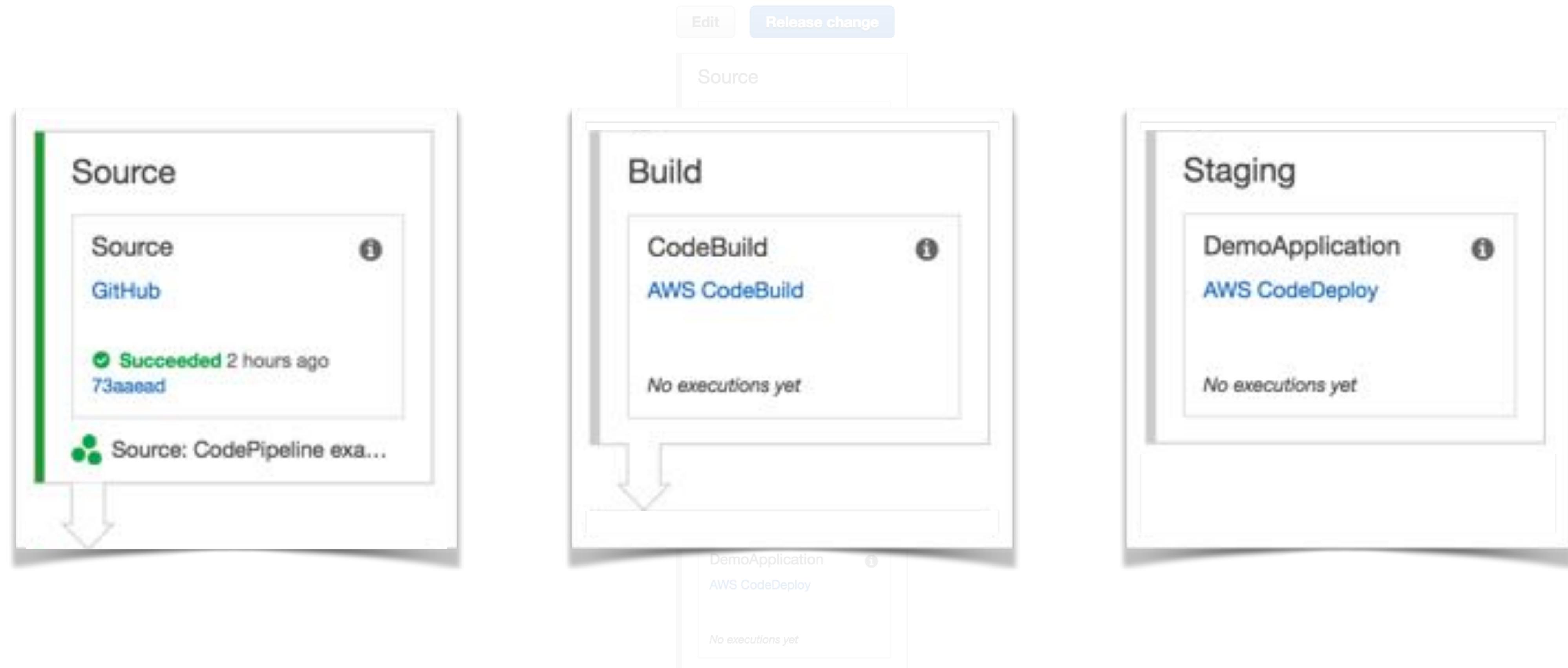


CodePipeline



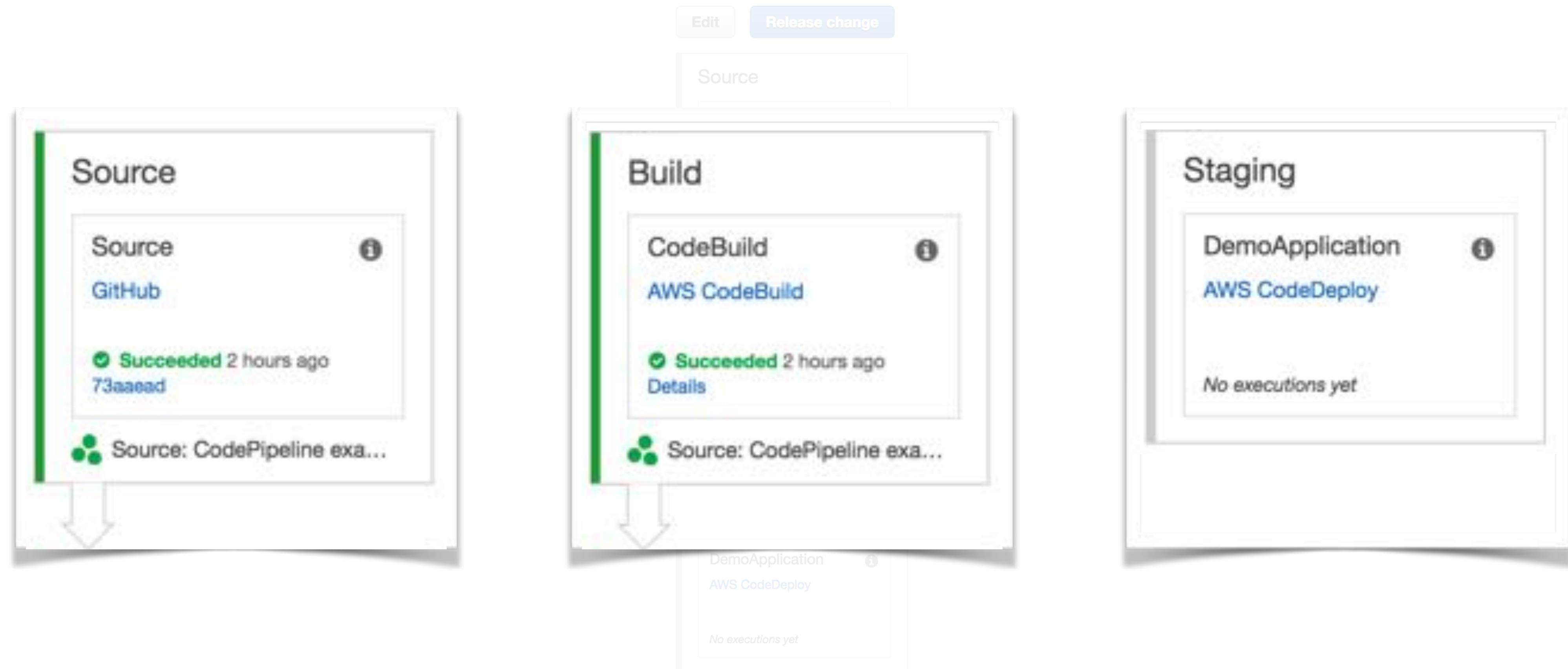


CodePipeline



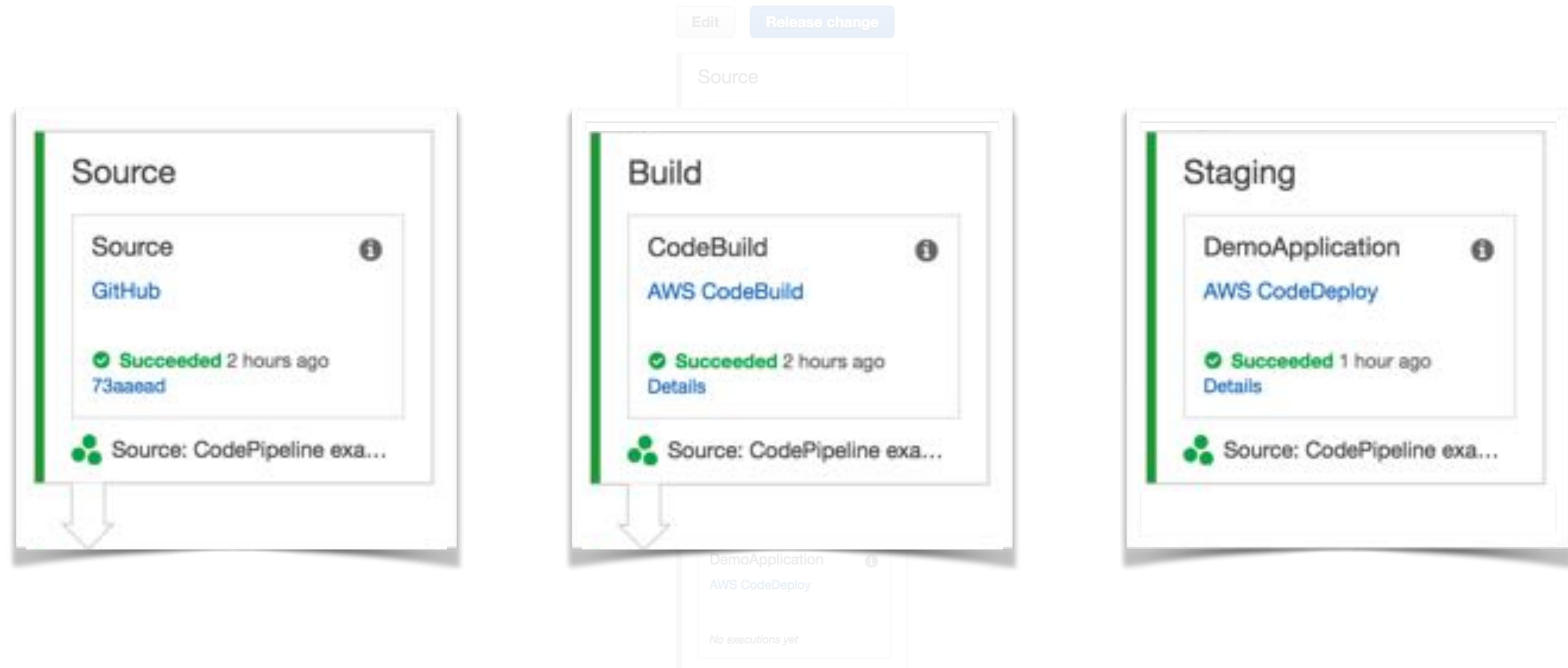


CodePipeline





CodePipeline



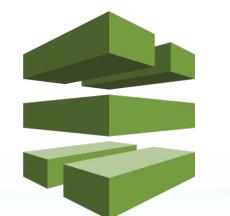
Other actions

CodePipeline - Other actions - Test



- Jenkins
- AWS CodeBuild
- BlazeMeter
- Ghost Inspector UI Testing
- HPE StormRunner Load
- Nouvola
- Runscope API Monitoring

CodePipeline - Other actions - Approval



SNS topic ARN

arn:aws:sns:us-east-1:012345678910:demo-a

URL for review

<https://staging.demo-application.com>

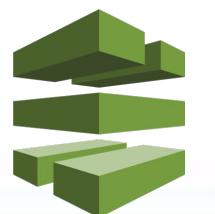
The URL you enter here will be provided to the reviewer as part of the approval request. It should begin with 'http://' or 'https://'.

Comments

Please test and approve this build!

The information you provide will be displayed to the approver in email notifications or the console.

CodePipeline - Other actions - Approval



SNS topic ARN

678910:demo-a

URL for review

on.com

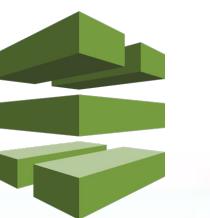
ided to the
uest. It should

Comments

build!

displayed to the
approver in email notifications or the console.

The screenshot shows a Lambda function named 'Approval' with the ARN '678910:demo-a'. The status is 'Manual approval' with a note 'Waiting for approval' for 4 hours ago. A blue 'Review' button is visible. Below the function name, there's a 'Source: CodePipeline exa...' link. The background of the slide has faint, semi-transparent text and arrows related to the Lambda function and its integration with CodePipeline.



CodePipeline - Other actions - Approval

SNS topic ARN

Approval

Approval

Manual approval

⚠ Waiting for approval 4 hours ago

Review

Source: CodePipeline exa...

view

ments

Approval

Manual approval

✓ Approved just now

Details

Source: CodePipeline exa...

678910:demo-a

on.com

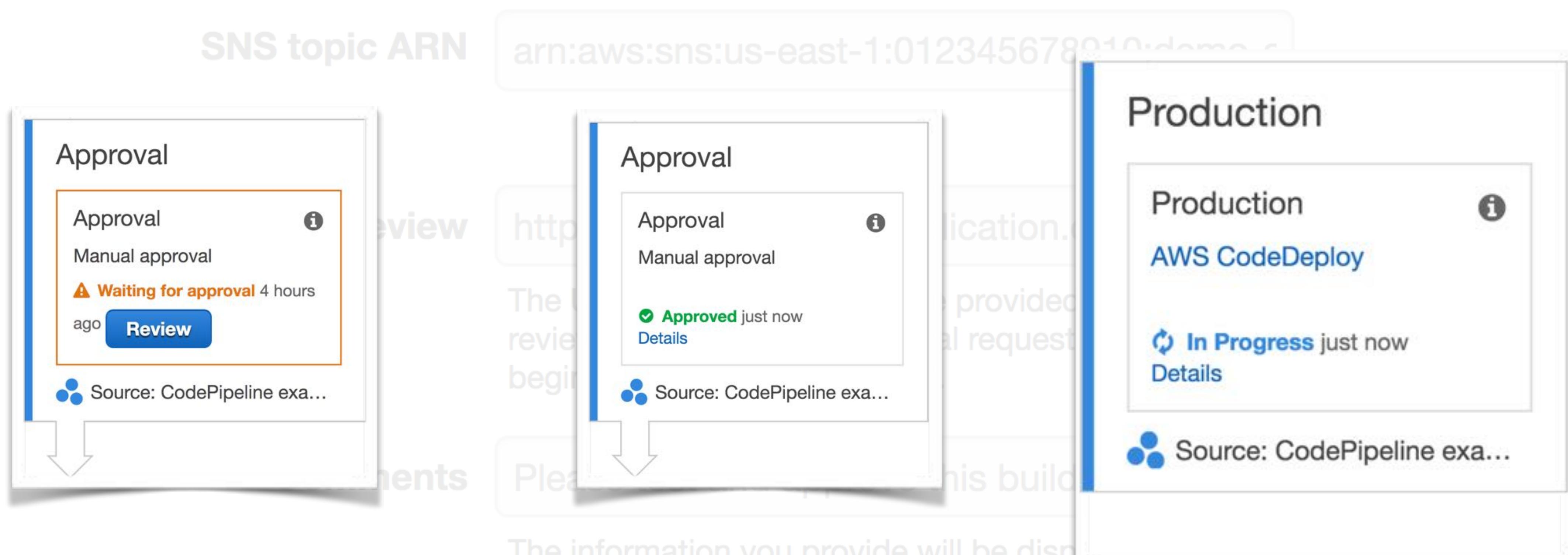
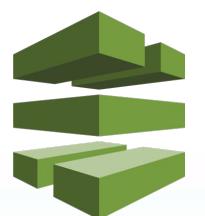
ided to the quest. It should

build!

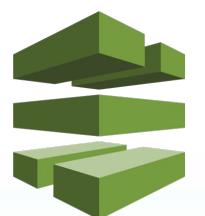
displayed to the approver in email notifications or the console.

The screenshot shows two views of an AWS Lambda function's configuration. On the left, a preview view shows a step named 'Approval' with a status of 'Waiting for approval 4 hours ago'. A blue 'Review' button is visible. On the right, the full configuration view shows the same step, but the status is now 'Approved just now'. The status is indicated by a green checkmark icon and the word 'Approved' in green text. Below the status, there is a 'Details' link. Both views include a 'Source: CodePipeline exa...' message at the bottom. The background of the slide contains faint, semi-transparent text related to SNS topics and Lambda functions.

CodePipeline - Other actions - Approval



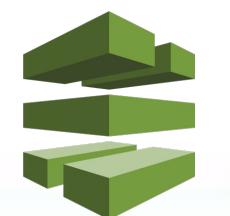
CodePipeline - Other actions - Approval



SNS topic ARN `arn:aws:sns:us-east-1:012345678910:demo-a`

The information you provide will be displayed to the approver in email notifications or the console.

CodePipeline - Other actions - Approval



SNS topic ARN

arn:aws:sns:us-east-1:012345678910:demo-a

URL for review

<https://staging.demo-application.com>

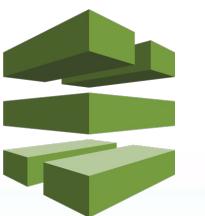
The URL you enter here will be provided to the reviewer as part of the approval request. It should begin with 'http://' or 'https://'.

Comments

Please test and approve this build!

The information you provide will be displayed to the approver in email notifications or the console.

CodePipeline - Other actions - Invoke



Function name*

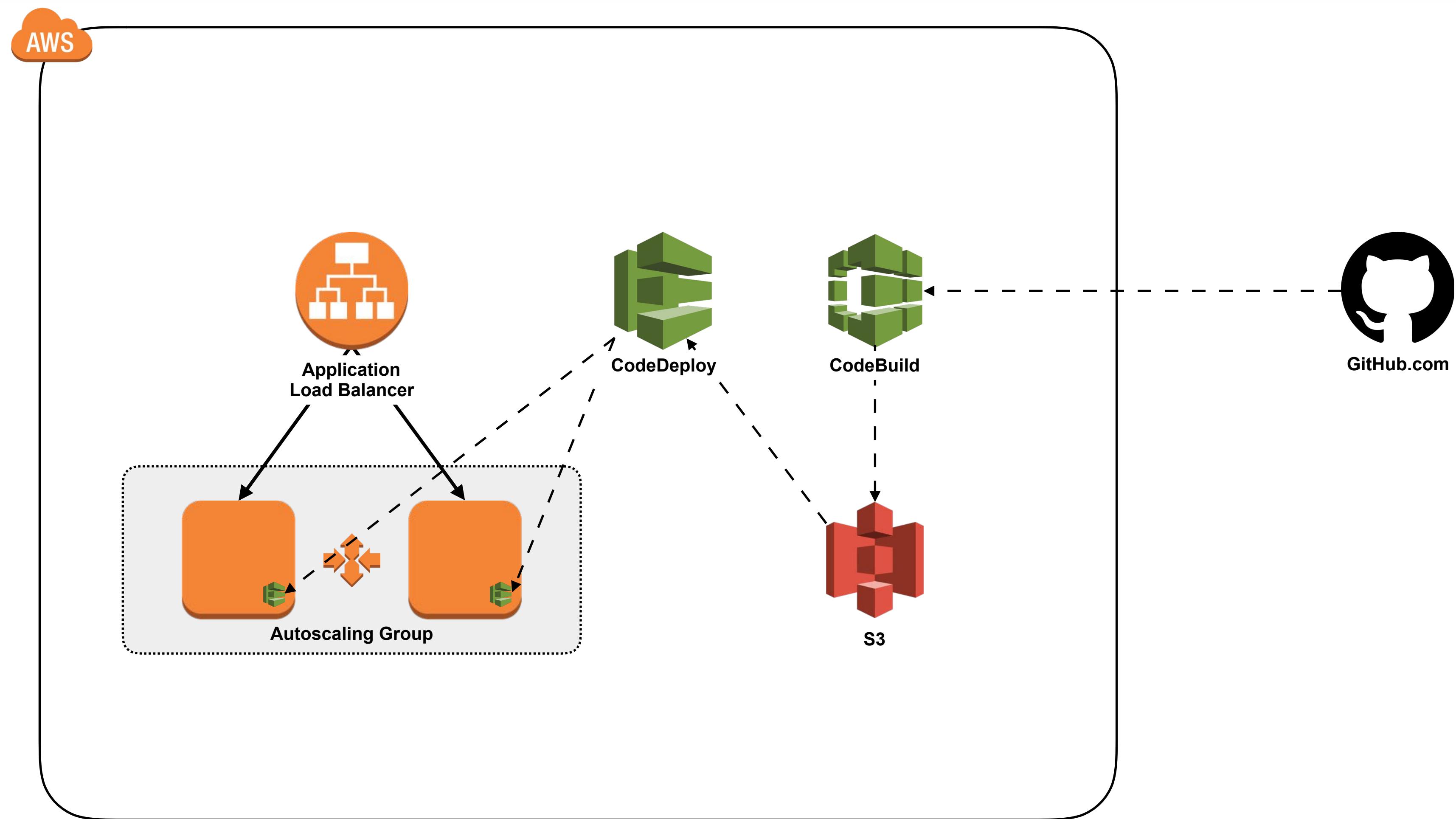
DoSomethingFancy

User parameters

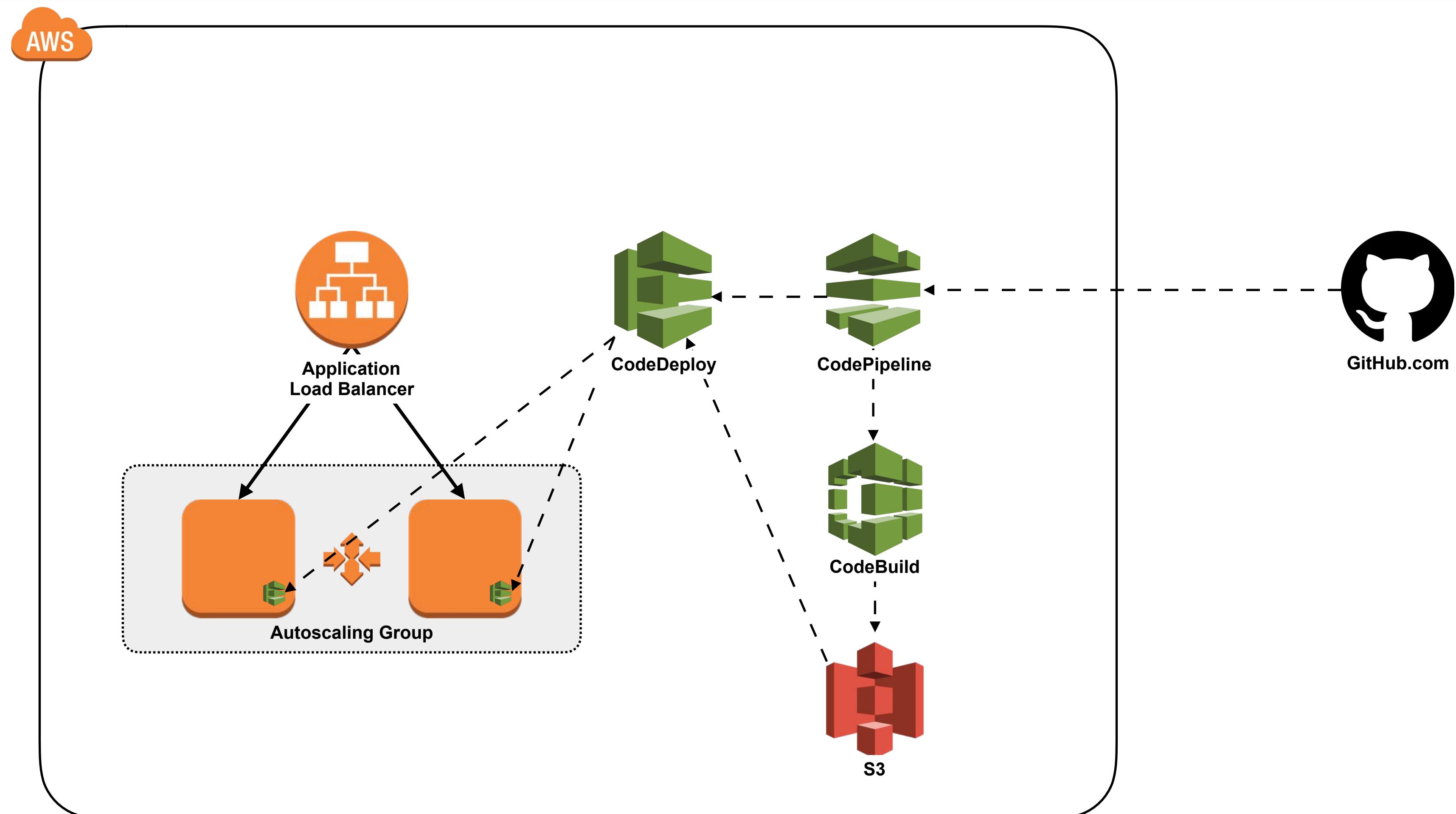
{"foo":"bar"}

Putting it all together

CodePipeline → CodeBuild → CodeDeploy



CodePipeline → CodeBuild → CodeDeploy



Pull Requests

CodeBuild - Pull Requests

Webhook



Rebuild every time a code change is pushed to this repository

Branch filter

master



CodeBuild - Pull Requests

 All checks have passed [Hide all checks](#)

1 successful check

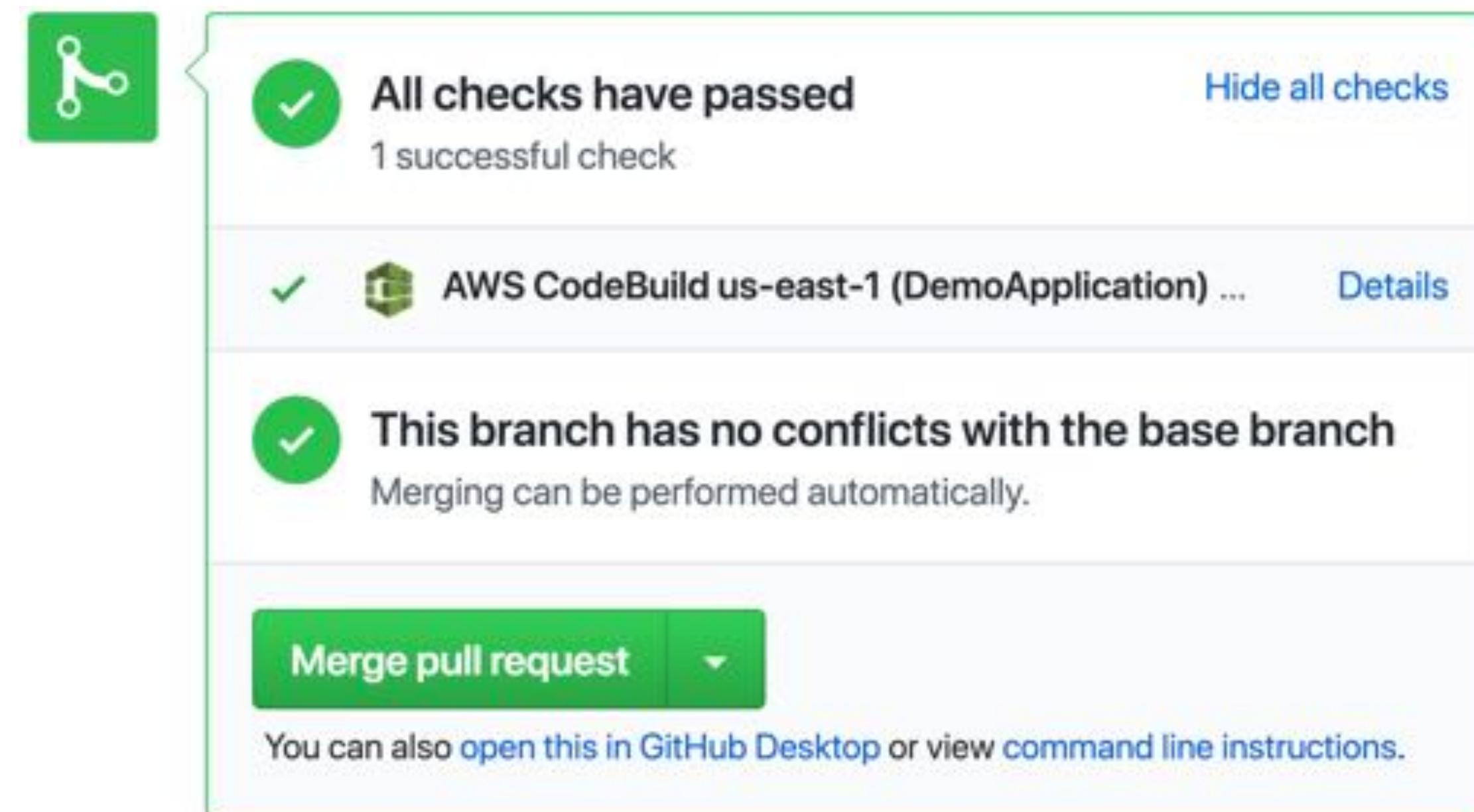
	 AWS CodeBuild us-east-1 (DemoApplication) ... Details
---------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

 This branch has no conflicts with the base branch
Merging can be performed automatically.

[Merge pull request](#) ▾

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

CodeBuild - Pull Requests



<https://docs.aws.amazon.com/codebuild/latest/userguide/sample-github-pull-request.html>

Closing thoughts

<https://joind.in/talk/e033f>

Thanks!

<https://joind.in/talk/e033f>