



# aws INNOVATE

MODERN APPLICATIONS EDITION

27 & 28 October 2021

# Developing container apps to building release pipeline with AWS Copilot

Donnie Prakoso

Senior Developer Advocate, ASEAN  
Amazon Web Services



@donnieprakoso



donnieprakoso



[go.donnie.id/youtube](https://go.donnie.id/youtube)



donnieprakoso

# Agenda layout

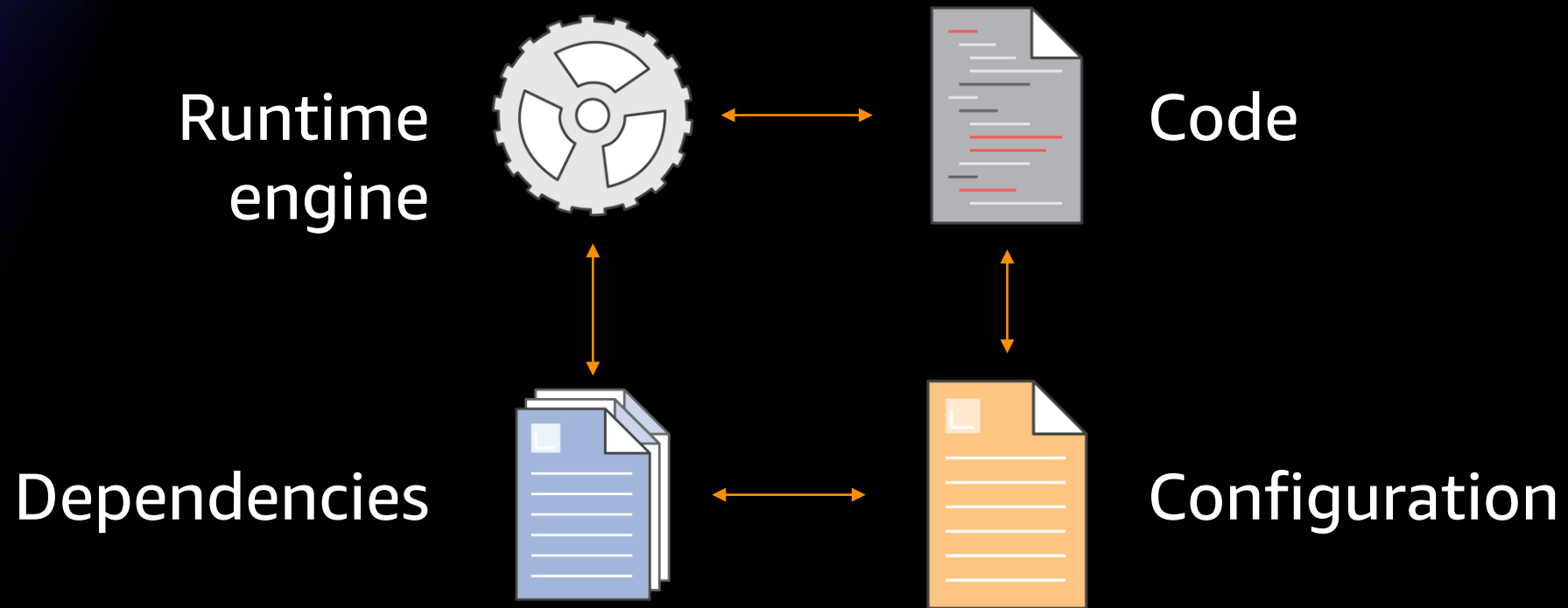
- Orchestrating containers with Amazon Elastic Container Service (Amazon ECS) using AWS Fargate
- Deploying containerized apps with AWS Copilot
- CI/CD foundations
- Demo: Implementing CI/CD with AWS Copilot

# Key takeaways

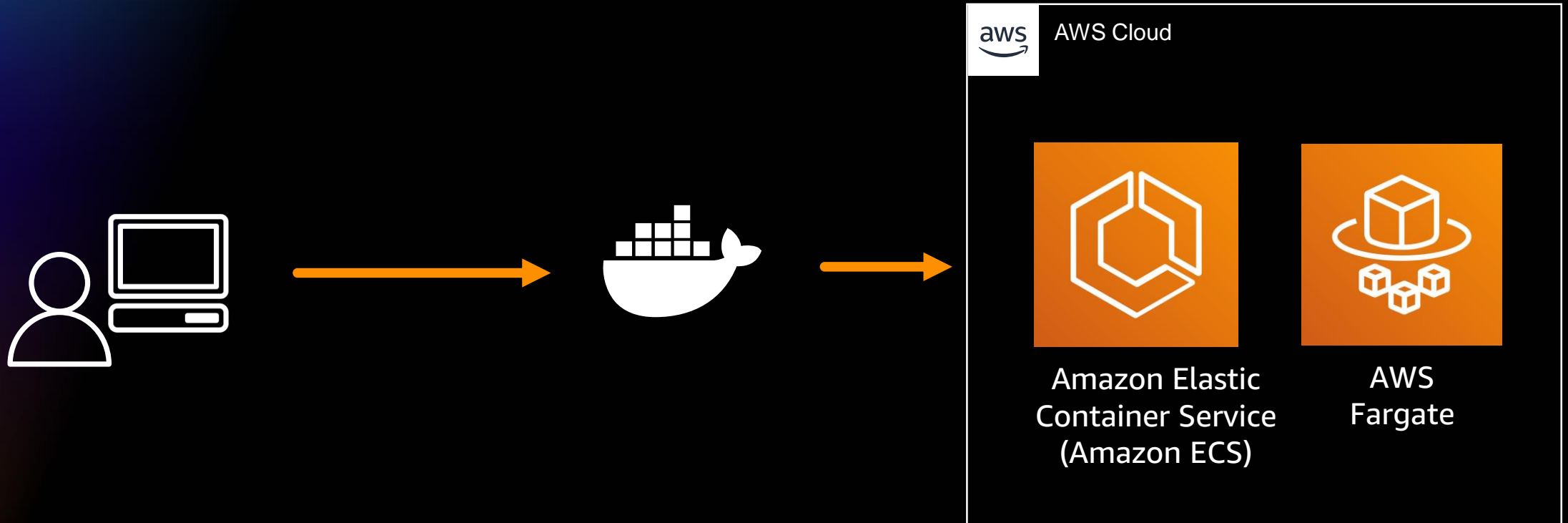
- How to ship your containerized apps into Amazon ECS and AWS Fargate using AWS Copilot
- How to create multiple environments using AWS Copilot
- How to implement CI/CD for a single or multiple environments
- How to do implement integrate testing
- How to separate configuration for multiple environments
- How to have fun with AWS Copilot

# Developing with containers

# Application environment components



# Typical process to ship application



# Amazon Elastic Container Service (Amazon ECS)



Container-level networking



Advanced task placement



Deep integration with AWS platform



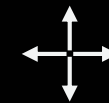
Amazon ECS command-line interface (CLI)



Global footprint



Powerful scheduling engines



Automatic scaling

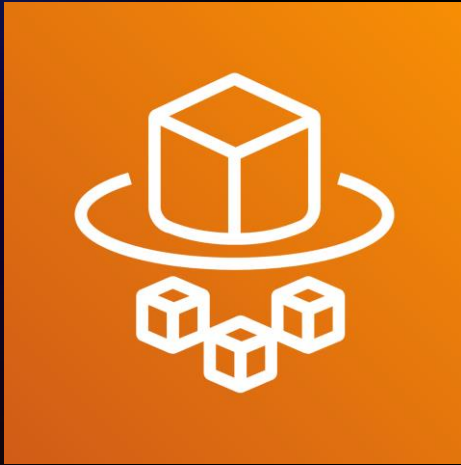


Amazon CloudWatch metrics



Load balancers

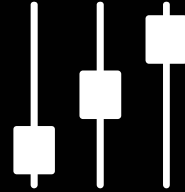
# AWS Fargate



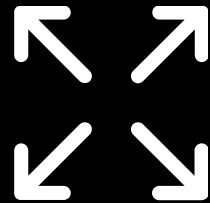
**AWS Fargate**  
Containers on demand



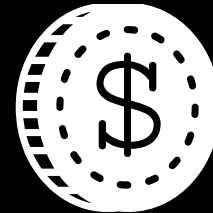
No  
infrastructure



Manage everything at  
container level

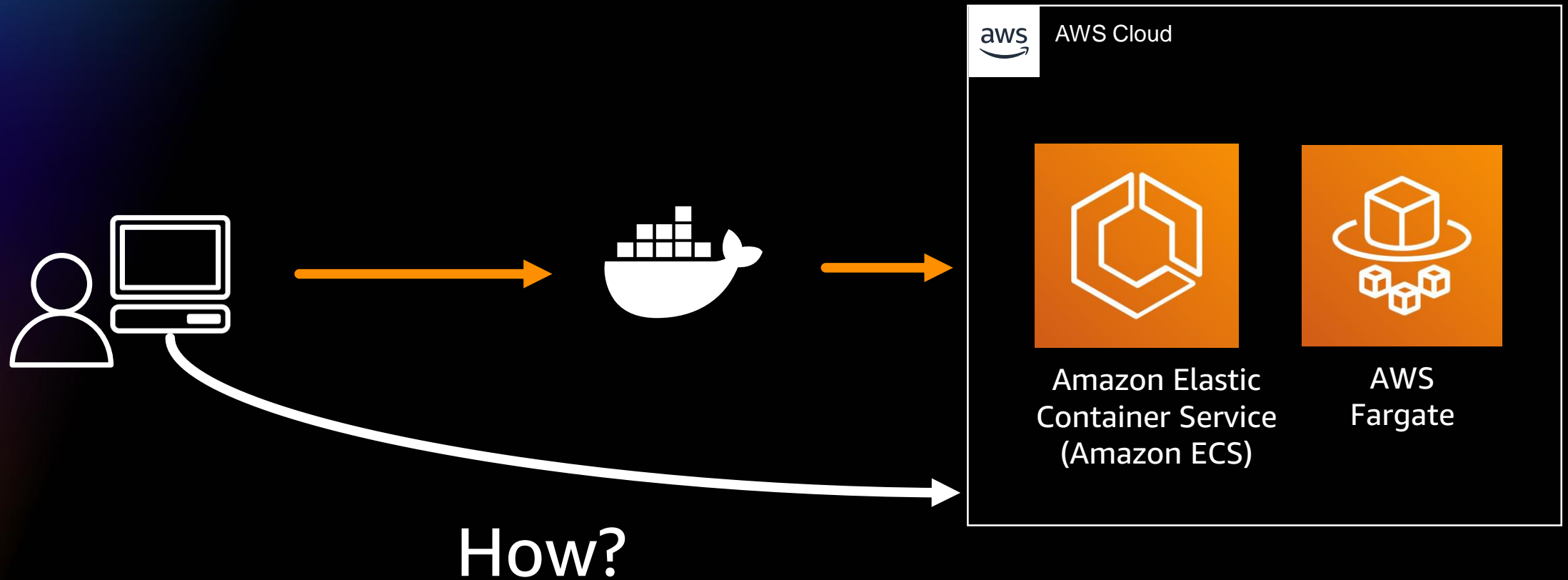


Launch quickly &  
scale easily



Resource-based  
pricing

# Typical process to ship application






# Container deployment challenges


- How do I deploy applications?
- How to check the applications?
- How do I add a service & integrate with AWS services?
- How can I test without affecting productions?
- How do I release applications?
- How about cleaning up?

# Introducing AWS Copilot Command Line Interface (CLI)

<https://aws.github.io/copilot-cli>

 AWS Copilot CLI

  Search

 aws/copilot-cli  
v1.6.0 ☆1.4k 🗨159

## Your toolkit for containerized applications on AWS

AWS Copilot is an open source command line interface that makes it easy for developers to **build**, **release**, and **operate** production ready containerized applications on Amazon ECS and AWS Fargate.

[Get started →](#)

AWS Copilot CLI

```
File: copilot/frontend/manifest.yml
1 # The manifest for the "frontend" service.
2
3 # Your service name will be used in naming your resources like log
4 # groups, ECS services, etc.
5 name: frontend
6 # The "architecture" of the service you're running.
7 type: Load Balanced Web Service
8
9 image:
10 # Docker build arguments. You can specify additional overrides h
11 ere.
12 build: frontend/Dockerfile
13 # Port exposed through your container to route traffic to it.
14 port: 8080
15
16 http:
17 # Requests to this path will be forwarded to your service.
18 # To match all requests you can use the "/" path.
19 path: '/'
20 # You can specify a custom health check path. The default is "/"
21 # healthcheck: '/'
22
23 # Number of CPU units for the task.
24 cpu: 256
25 # Amount of memory in MiB used by the task.
```

```
~/D/p/my-app >>> copilot pipeline status
Pipeline Status
Stage      Transition      Status
Source
└─ SourceCodeFor-emoji-race      ENABLED      Succeeded
Build
└─ Build                        ENABLED      Succeeded
DeployTo-test
└─ CreateOrUpdate-tracker-test  ENABLED      Succeeded
└─ TestCommands                  Succeeded

Last Deployment
Updated At   1 week ago
~/D/p/my-app >>>

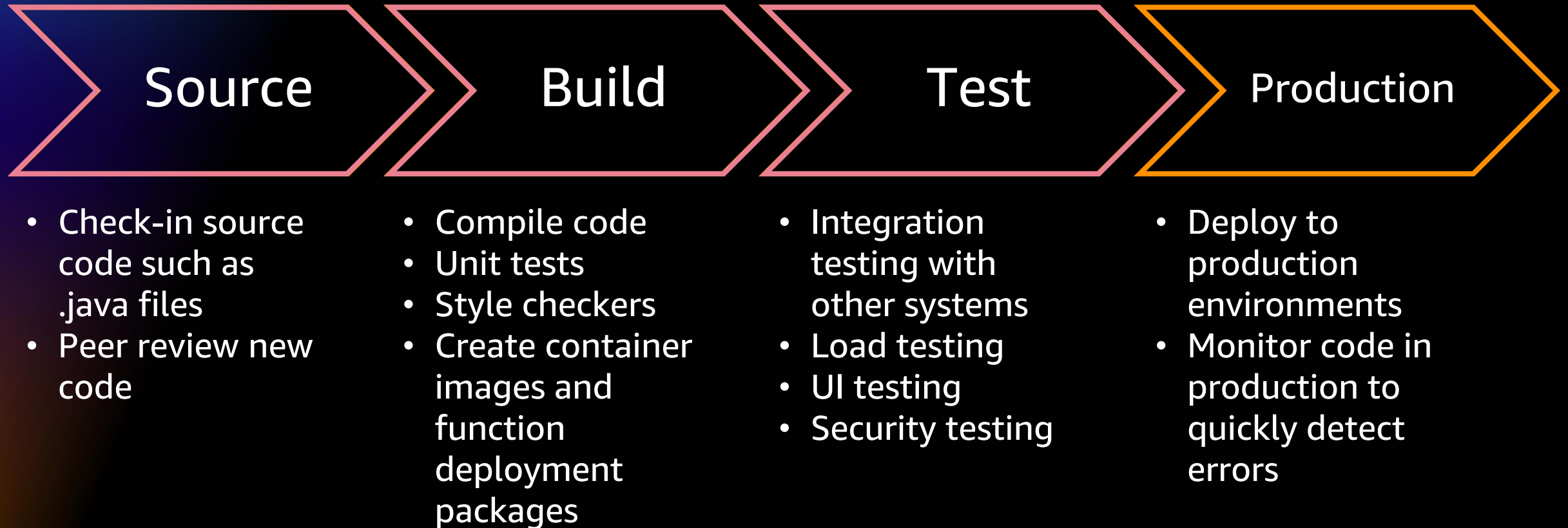
~/D/p/my-app >>> copilot svc status
Only found one deployed service tracker in environment test
Service Status
ACTIVE 1 / 1 running tasks (0 pending)

Last Deployment
Updated At   1 week ago
```

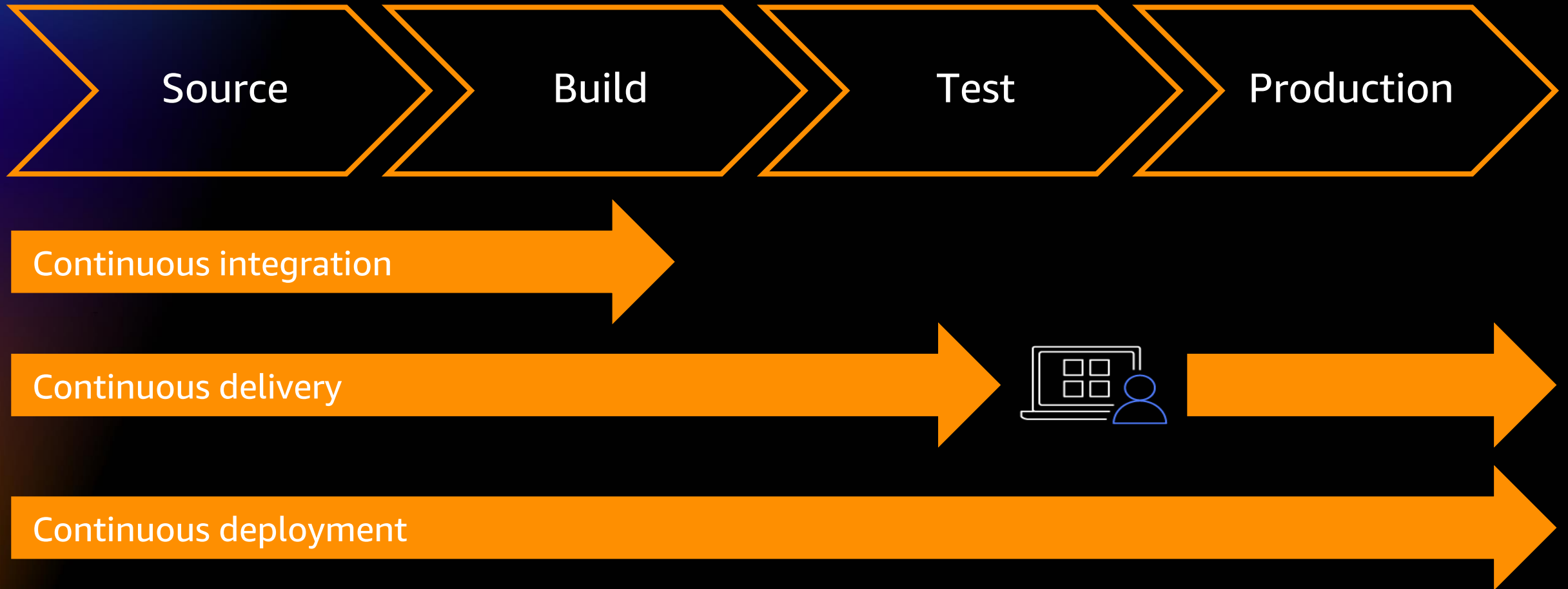
# Demo 1

# CI/CD foundations

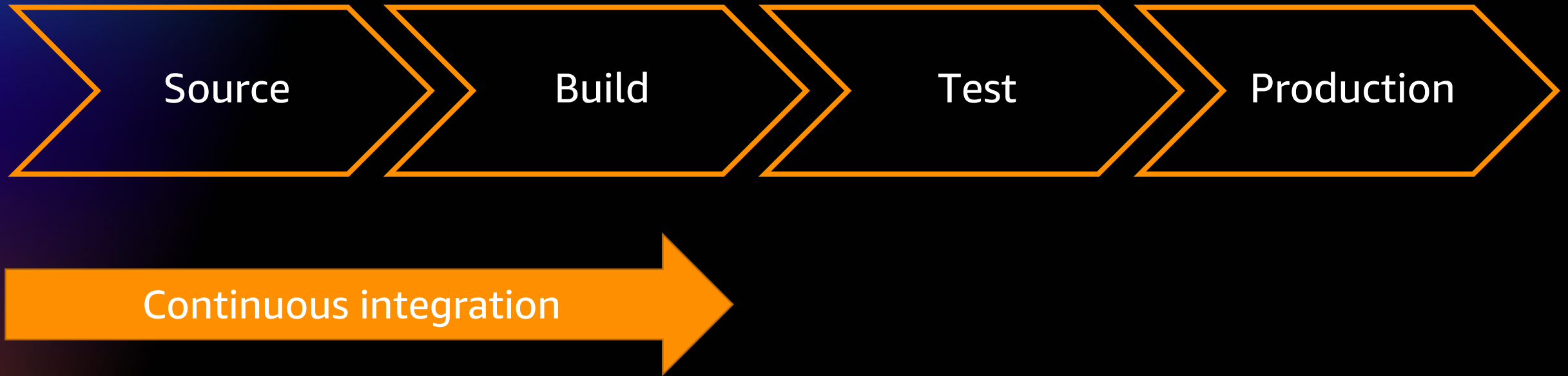
# Release process stages



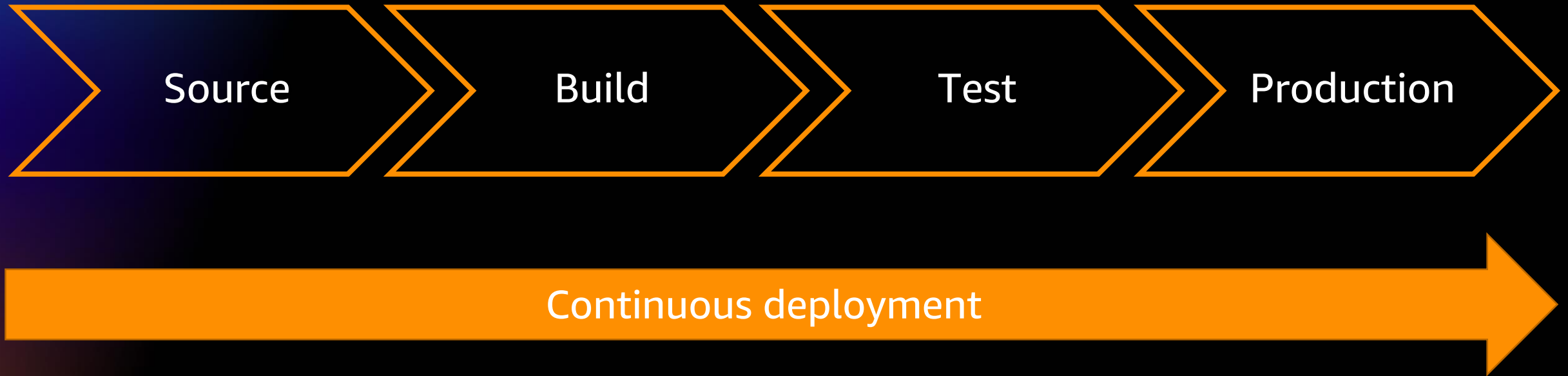
# Release process stages



# Continuous integration goals



# Continuous deployment goals

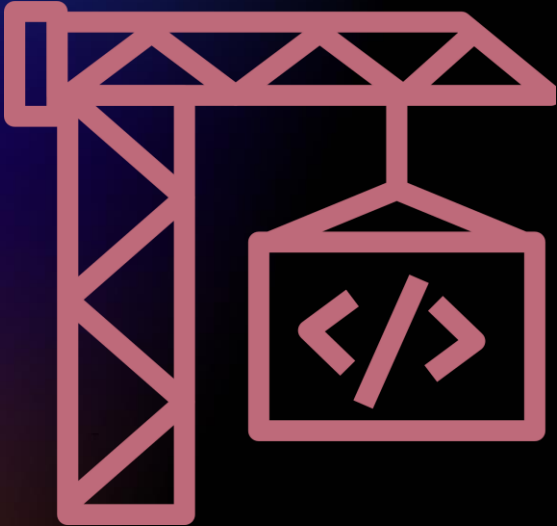


# AWS CodePipeline



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

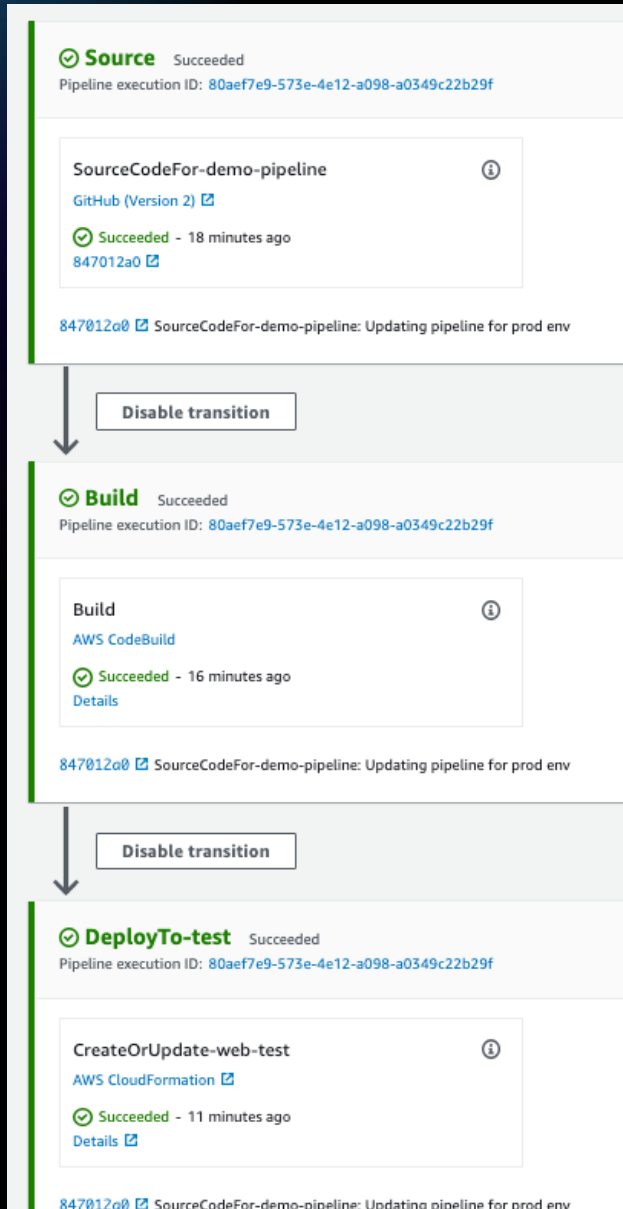
# AWS CodeBuild



- Each build runs in a new Docker container for a consistent, immutable environment
- Docker and AWS Command Line Interface (AWS CLI) are installed in every official AWS CodeBuild image
- Provide custom build environments suited to your needs through the use of Docker images

# Building release pipeline with AWS Copilot

# Release diagram for a single environment



\$ copilot pipeline init

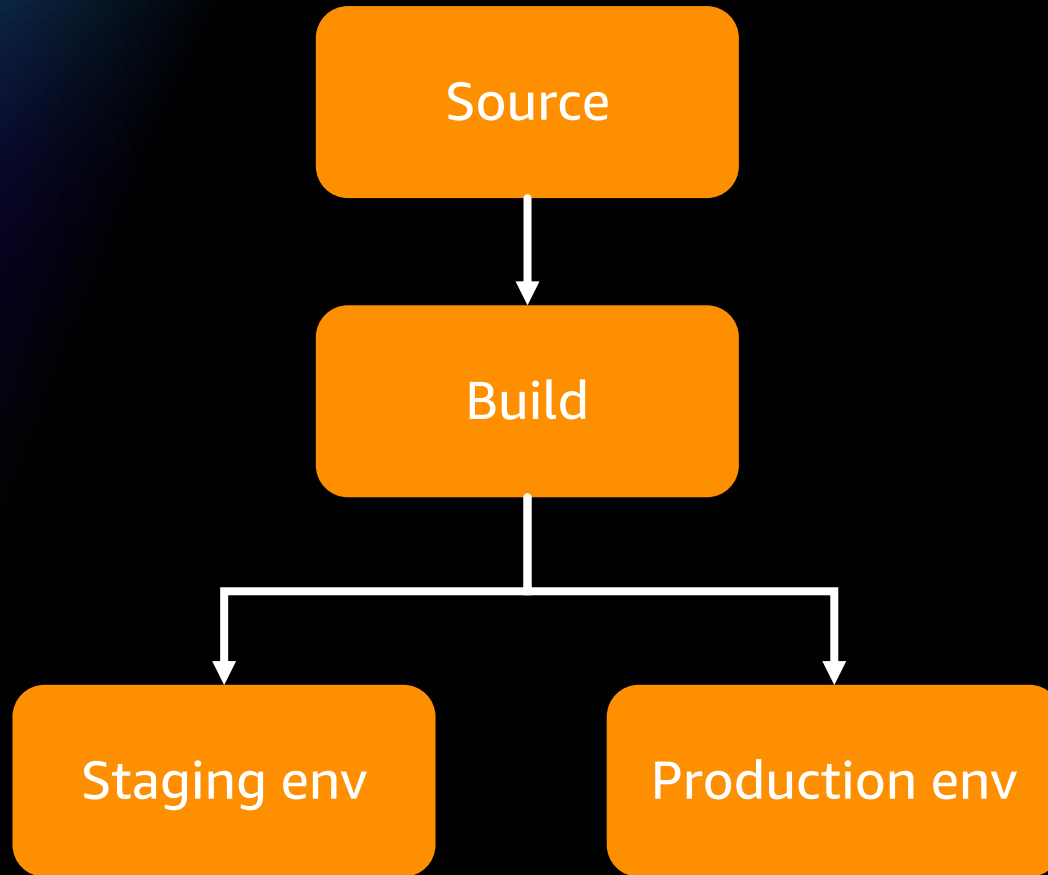
buildspec.yaml

pipeline.yaml

\$ copilot pipeline update

# Demo 2

# Release diagram for multiple environments



\$ copilot env init

\$ copilot pipeline init

\$ copilot pipeline update

# Demo 3

# Adding testing for release pipeline

```
1 name: pipeline-with-test
2 version: 1
3 source:
4   provider: GitHub
5   properties:
6     branch: main
7     repository: <REPOSITORY_URL>
8
9 stages:
10   -
11     name: test
12     test_commands:
13       - pip install pytest
14       - pytest -v
```

pipeline.yaml

```
|
|test-commands:
| - command_1
| - command_2
```

\$ copilot pipeline update

# Demo 4

# Visit the Modern Applications Resource Hub for more resources

Dive deeper with these resources to help you develop an effective plan for your modernization journey.

- Build modern applications on AWS e-book
- Build mobile and web apps faster e-book
- Modernize today with containers on AWS e-book
- Adopting a modern Dev+Ops model e-book
- Modern apps need modern ops e-book
- Determining the total cost of ownership: Comparing Serverless and Server-based technologies paper
- Continuous learning, continuous modernization e-book
- ... and more!



<https://bit.ly/3yfOvbK>

**Visit resource hub »**

# AWS Training and Certification

Accelerate modernization with continuous learning



Free digital courses, including:  
[Architecting serverless solutions](#)  
[Getting started with DevOps on AWS](#)



Earn an industry-recognized credential:  
[AWS Certified Developer – Associate](#)  
[AWS Certified DevOps – Professional](#)



Hands-on classroom training  
(available virtually) including:  
[Running containers on Amazon Elastic  
Kubernetes Service \(Amazon EKS\)](#)  
[Advanced developing on AWS](#)



Create a self-paced learning roadmap  
[AWS ramp-up guide - Developer](#)  
[AWS ramp-up guide - DevOps](#)



Take [Developer](#)  
[and DevOps training](#)  
today



Learn more about  
[Modernization training](#) for you  
and your team

# Thank you for attending AWS Innovate Modern Applications Edition

We hope you found it interesting! A kind reminder to **complete the survey**.  
Let us know what you thought of today's event and how we can improve the event  
experience for you in the future.



[aws-apj-marketing@amazon.com](mailto:aws-apj-marketing@amazon.com)



[twitter.com/AWSCloud](https://twitter.com/AWSCloud)



[facebook.com/AmazonWebServices](https://facebook.com/AmazonWebServices)



[youtube.com/user/AmazonWebServices](https://youtube.com/user/AmazonWebServices)



[slideshare.net/AmazonWebServices](https://slideshare.net/AmazonWebServices)



[twitch.tv/aws](https://twitch.tv/aws)

# Thank you!

Donnie Prakoso  
Senior Developer Advocate, ASEAN  
Amazon Web Services



@donnieprakoso



donnieprakoso



[go.donnie.id/youtube](https://go.donnie.id/youtube)



donnieprakoso

