aws INNOVATE

MODERN APPLICATIONS EDITION

27&28 October 2021

Develop, operate, and integrate best practices for container apps with AWS Copilot

Donnie Prakoso

Sr. Developer Advocate, ASEAN
Amazon Web Services

@donnieprakoso

donnieprakoso

go.donnie.id/youtube

donnieprakoso

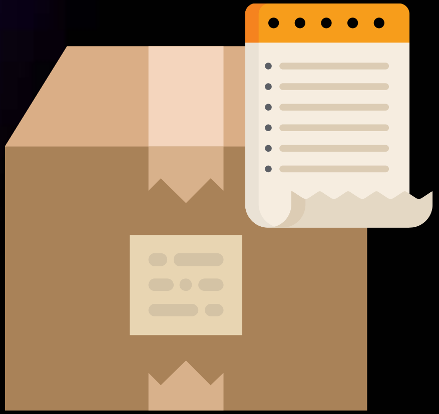
Agenda

- Best practices: Twelve-Factor App and design patterns
- Overview of Amazon Elastic Container Service (Amazon ECS) and AWS Fargate
- Build, release and operate containerized apps with AWS Copilot
- Demo

Key takeaways

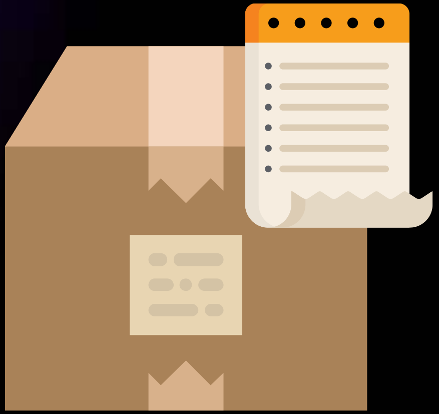
- How to use AWS Copilot to deploy your application
- How to define and use config and secrets
- How to implement backing service and service discovery
- How to implement CI/CD with AWS Copilot
- How to implement Pub/Sub architecture
- How to run one-time task

Twelve-Factor App principles



- I. Codebase
- II. Dependencies
- III. Config
- IV. Backing services
- V. Build, release, run
- VI. Processes
- VII. Port binding
- VIII. Concurrency
- IX. Disposability
- X. Dev/prod parity
- XI. Logs
- XII. Admin processes

Twelve-Factor App principles



I. Codebase

II. Dependencies

III. Config

IV. Backing services

V. Build, release, run

VI. Processes

VII. Port binding

VIII. Concurrency

IX. Disposability

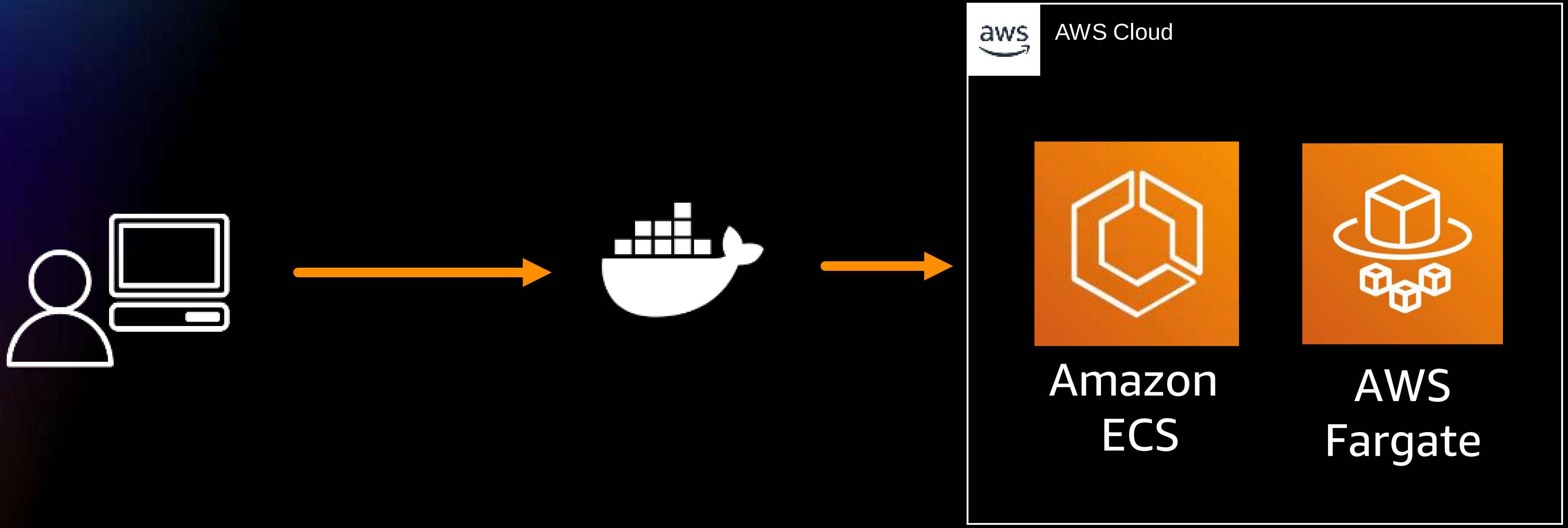
X. Dev/prod parity

XI. Logs

XII. Admin processes

Developing with containers

Typical process to ship app



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

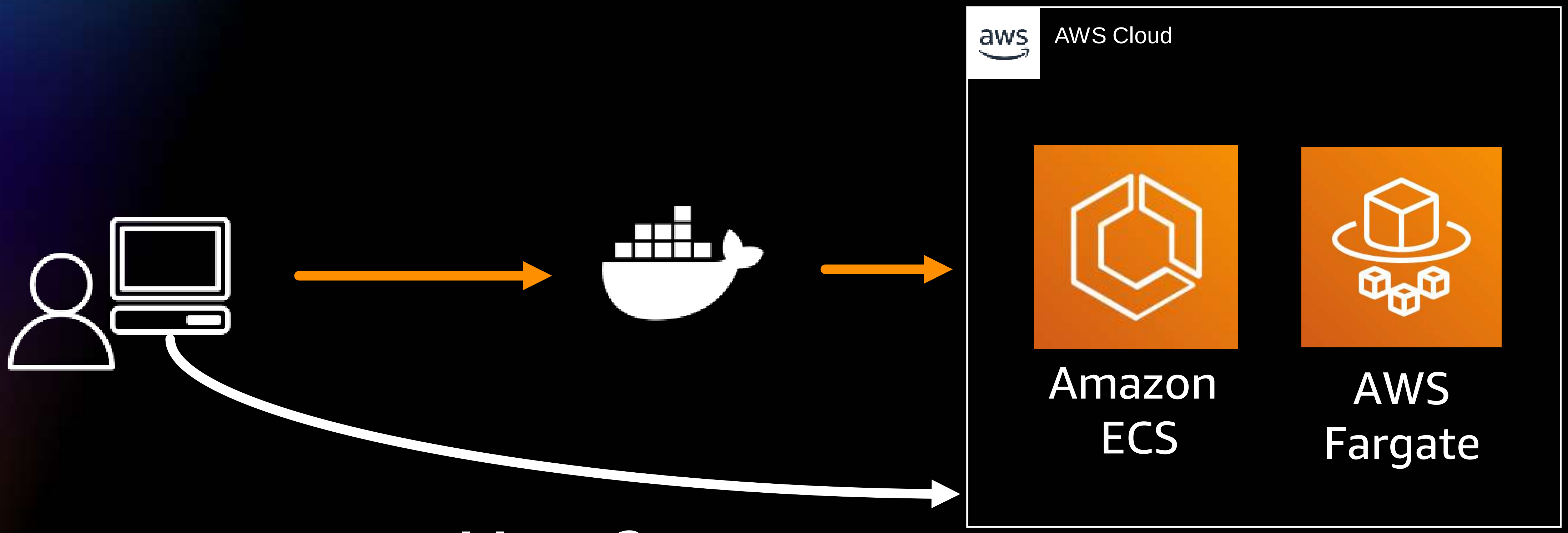
No infrastructure

Manage everything at container level

Launch quickly & scale easily

Resource-based pricing

Typical process to ship app



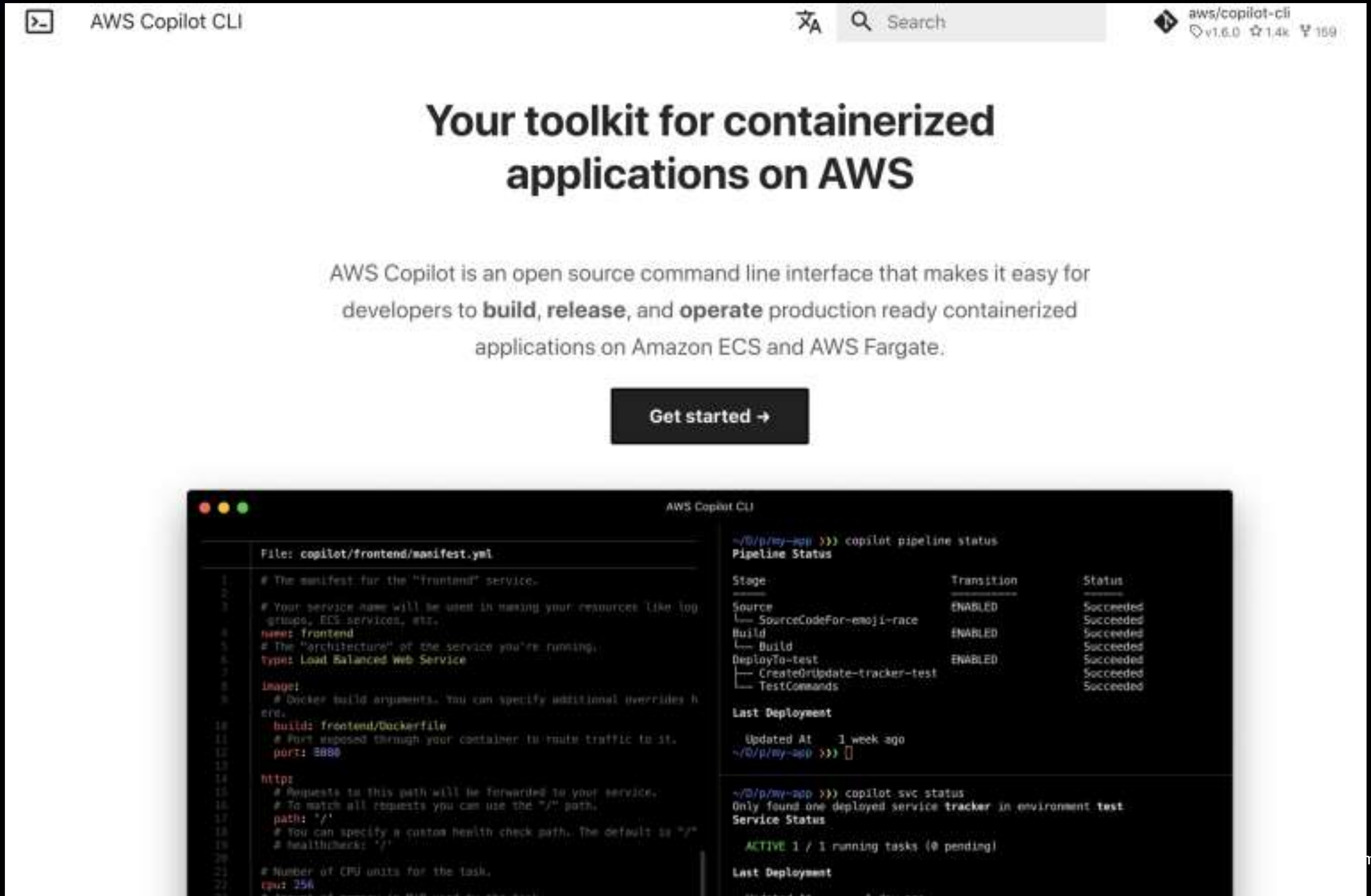
How?

Container deployment challenges

- How do I deploy applications?
- How do I add a service & integrate with AWS services?
- How can I test without affecting productions?
- How do I release applications?
- How to implement service discovery?

Introducing AWS Copilot command line interface (CLI)

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



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.

```
File: copilot/frontend/manifest.yml

# The manifest for the "frontend" service.

# Your service name will be used in naming your resources like log
 groups, ECS services, etc.
name: frontend
# The "architecture" of the service you're running.
type: Load Balanced Web Service

image:
  # Docker build arguments. You can specify additional overrides h
ere.
  build: frontend/Dockerfile
  # Port exposed through your container to route traffic to it.
  port: 8080

http:
  # Requests to this path will be forwarded to your service.
  # To match all requests you can use the "/" path.
  path: '/'
  # You can specify a custom health check path. The default is "/"
  # healthcheck: '/'

# Number of CPU units for the task.
cpu: 256
```

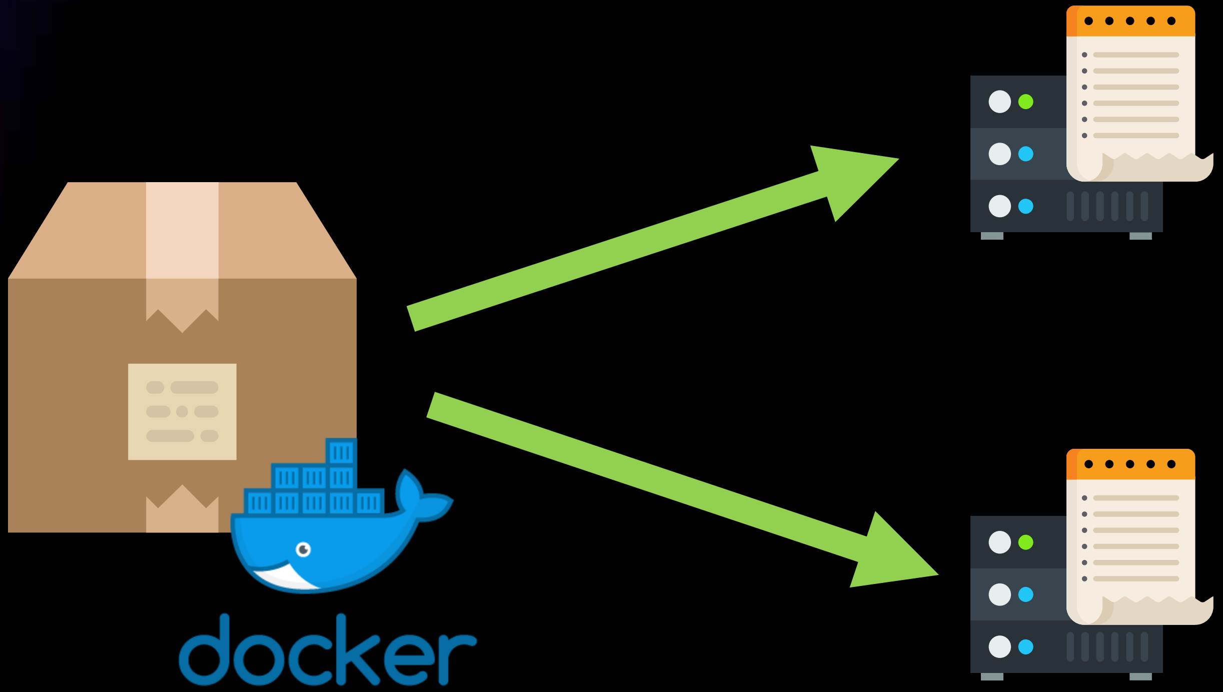
```
~/D/p/my-app ››› copilot pipeline status
Pipeline Status

Stage                               Transition        Status
-----                               ----------        ------
Source                              ENABLED           Succeeded
└── SourceCodeFor-emoji-race                          Succeeded
Build                               ENABLED           Succeeded
└── Build                                             Succeeded
DeployTo-test                       ENABLED           Succeeded
├── CreateOrUpdate-tracker-test                       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
```

Demo 1

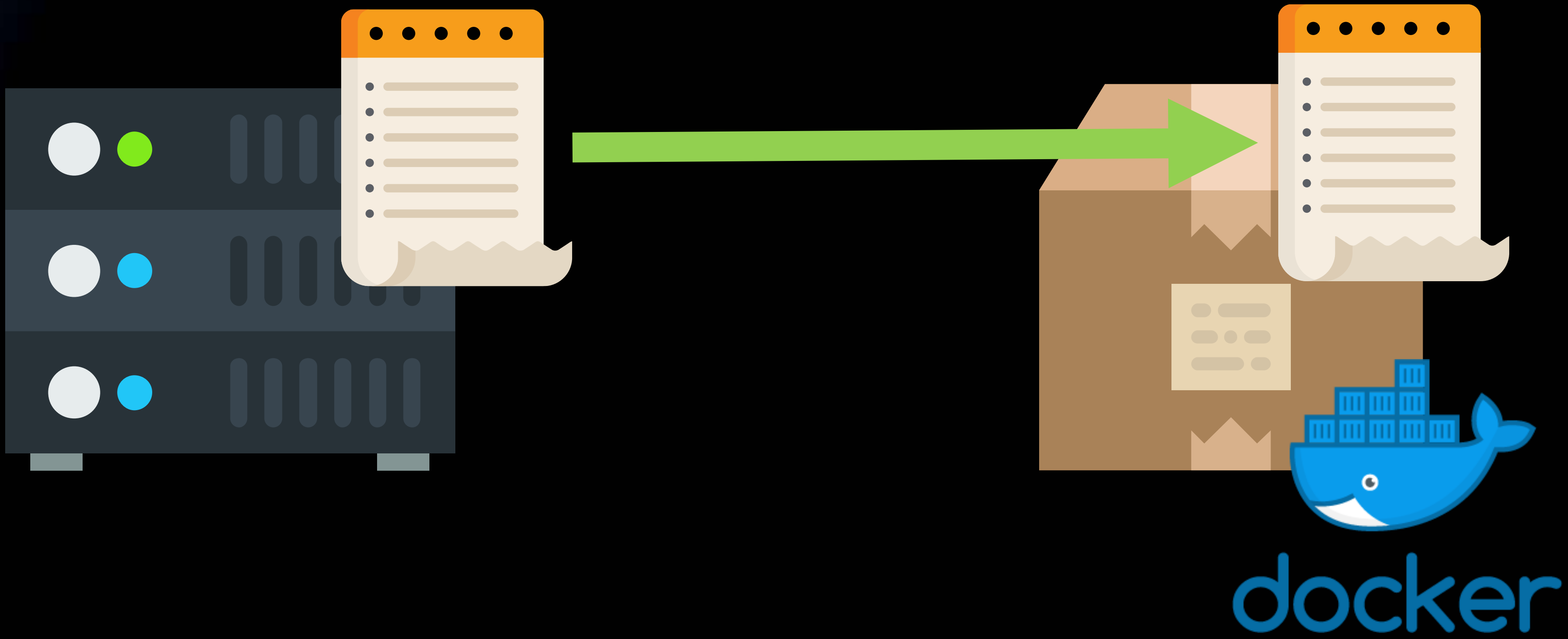
Twelve-Factor App: 03: Config

Same container deployed to both environments. Configuration is part of the environment on the host.



- Development
- Production

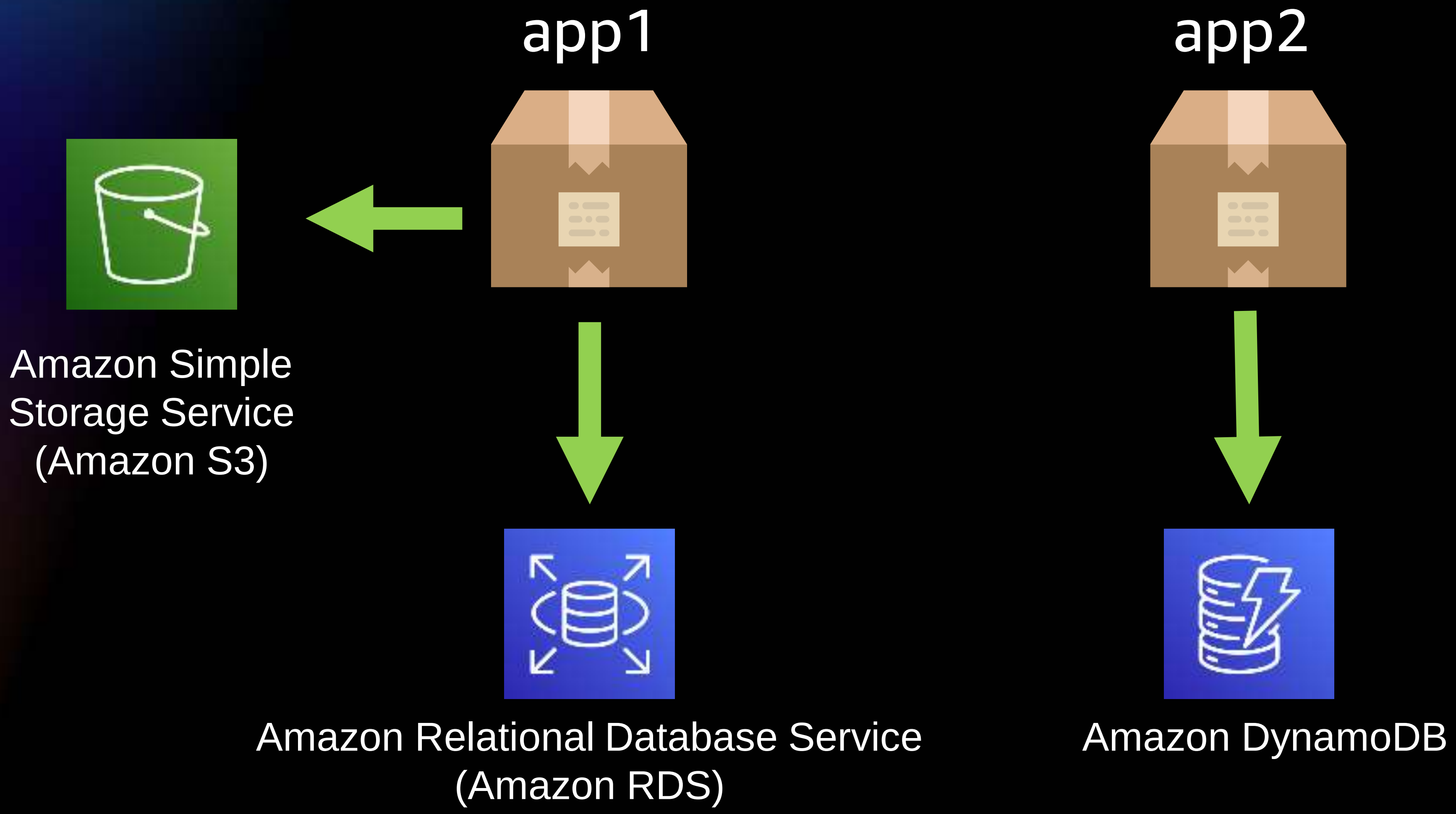
At runtime, the container gets config from the environment.



Demo 2

Twelve-Factor App: 04: Backing services

Treat local services just like remote third party ones



app1

app2

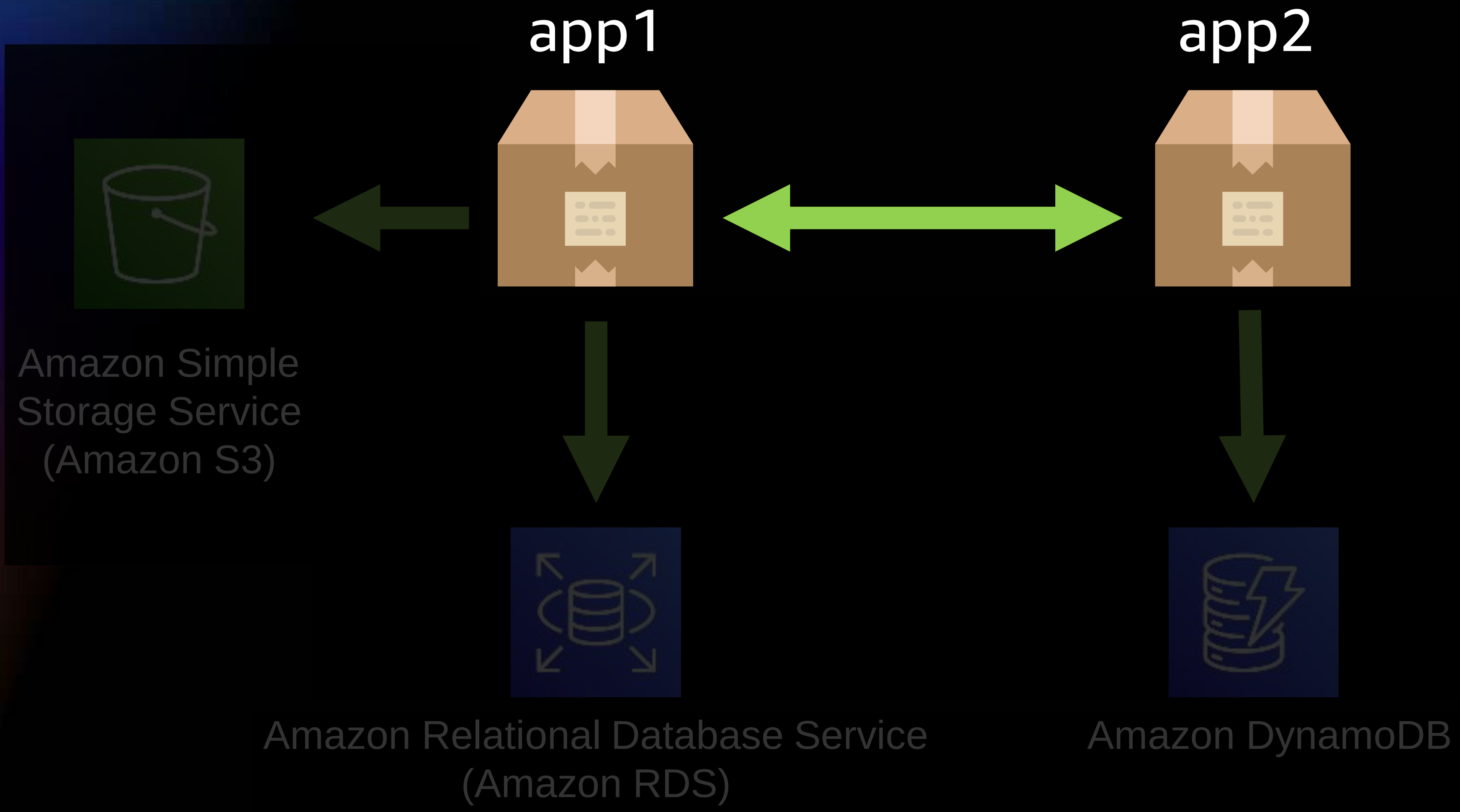
Amazon Simple Storage Service (Amazon S3)

Amazon Relational Database Service (Amazon RDS)

Amazon DynamoDB

Service discovery

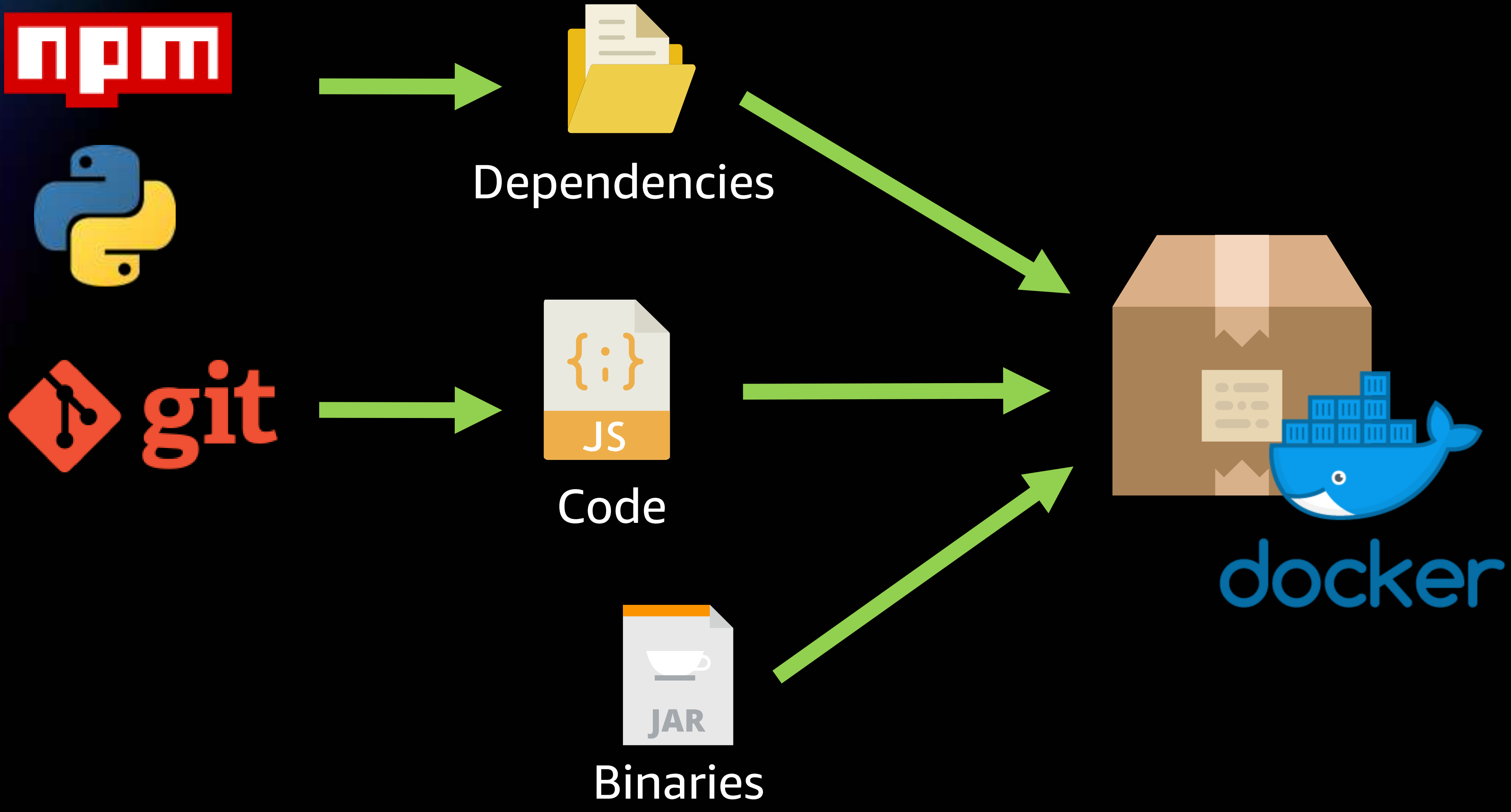
A mechanism for services to discover and interact with each other


app1
app2
Amazon Simple Storage Service (Amazon S3)
Amazon Relational Database Service (Amazon RDS)
Amazon DynamoDB

Demo 3

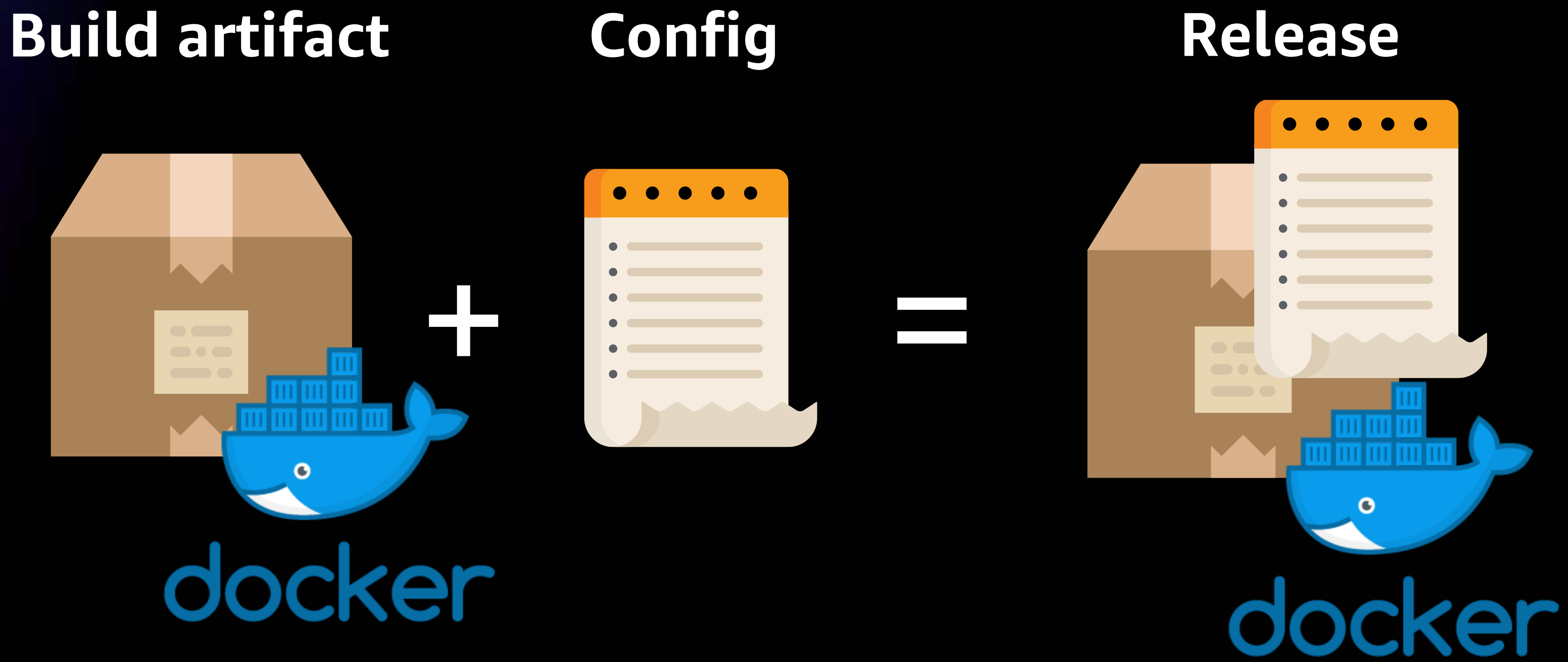
Twelve-Factor App:
05: Build, release, run

Build



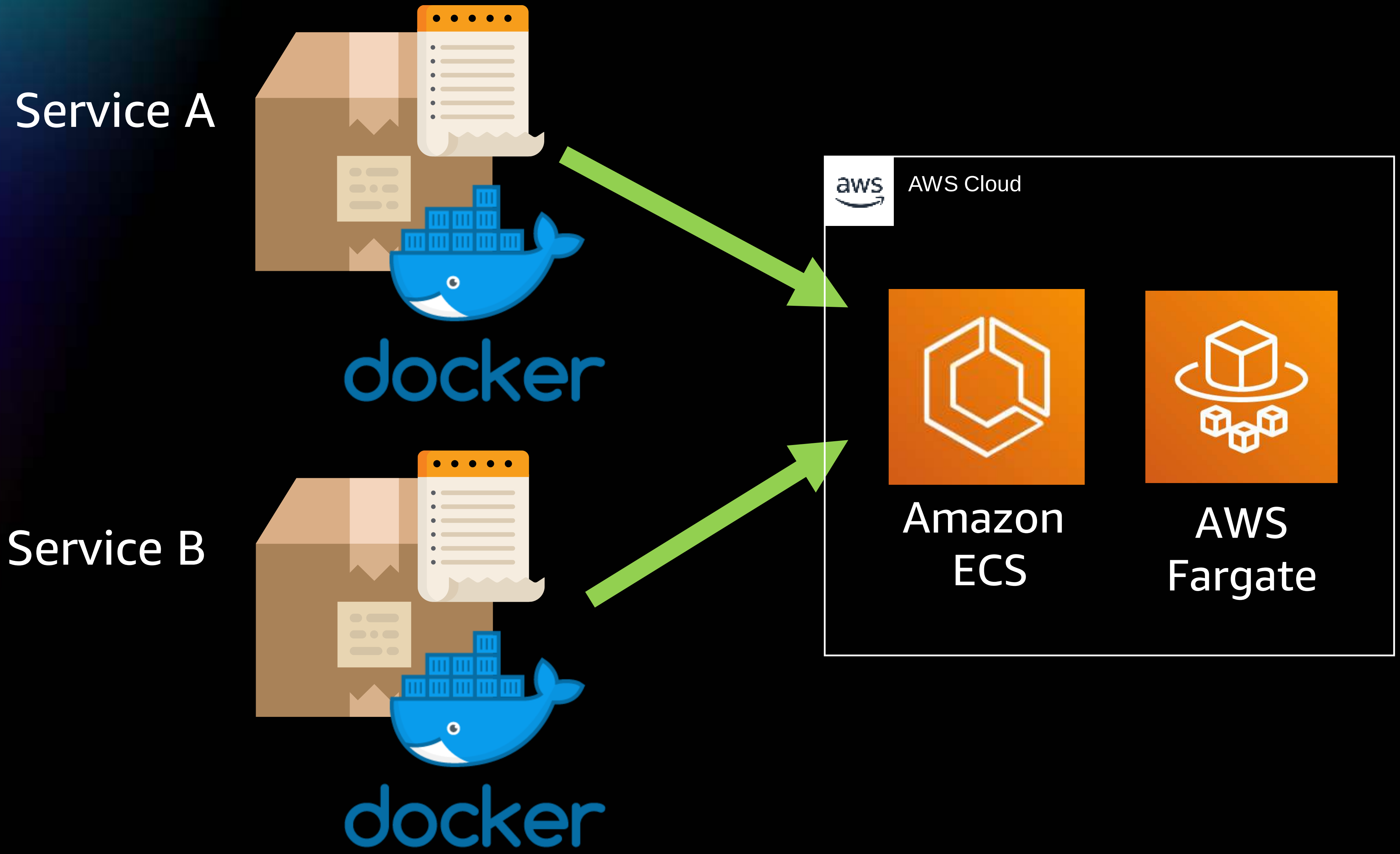
aws

Release



Build artifact + Config = Release

Run



Service A

Service B

AWS Cloud

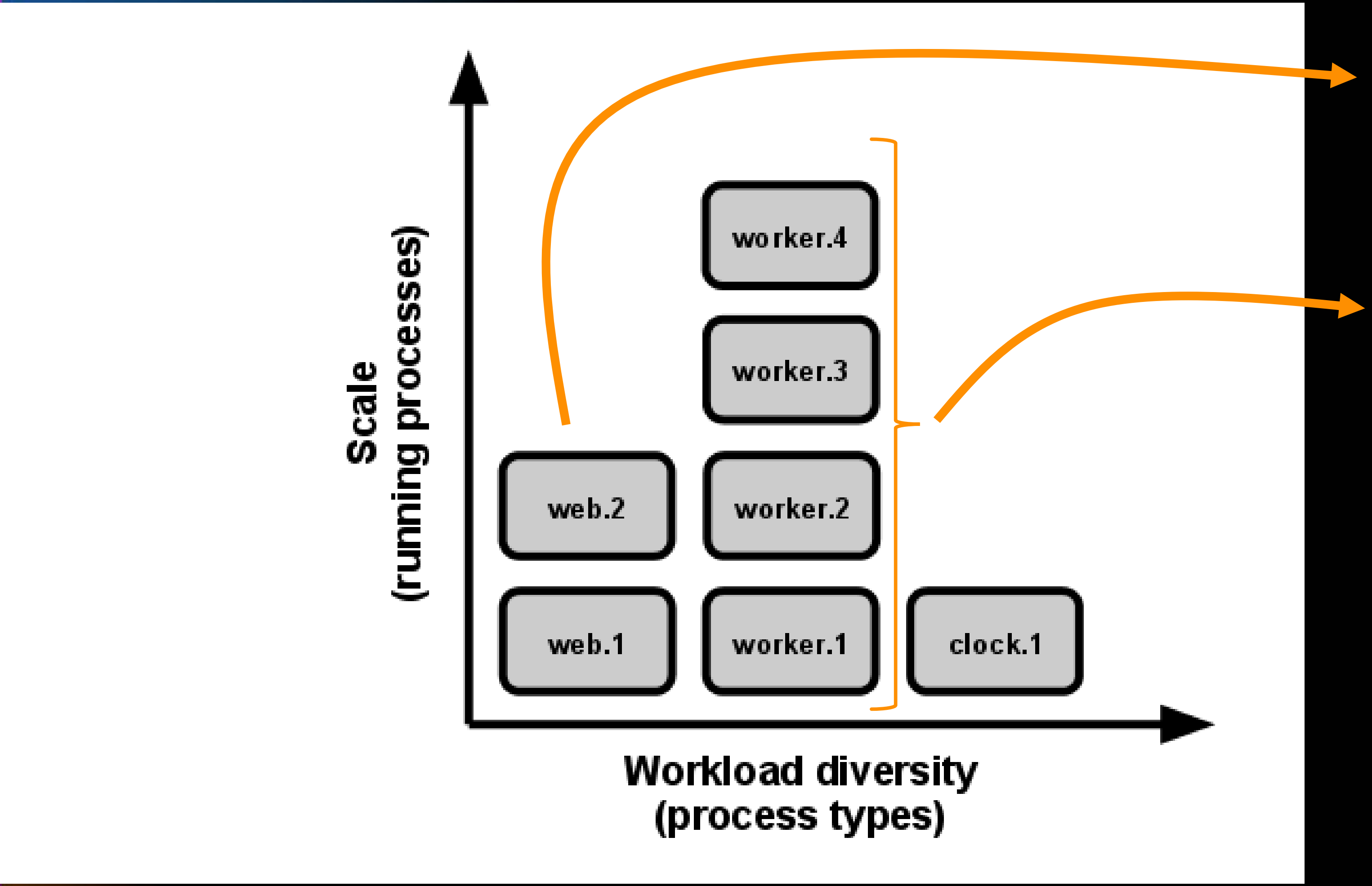
Amazon ECS

AWS Fargate

Demo 4

Twelve-Factor App: 08: Concurrency

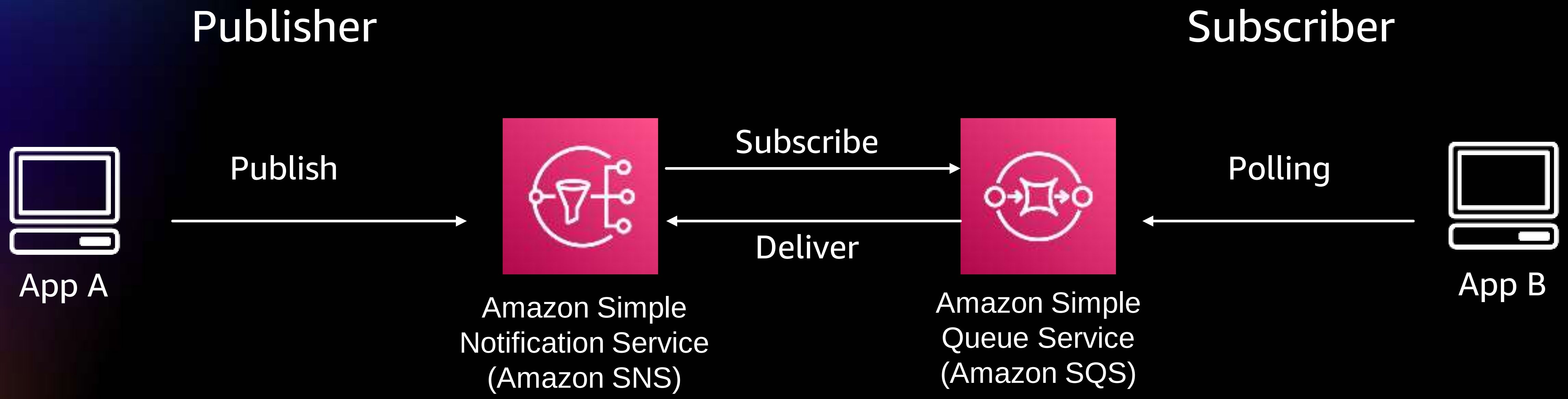
Concurrency



> Load balanced web service

> Worker service

Pub/Sub


Publisher
Subscriber
Publish
Subscribe
Deliver
Polling
App A
Amazon Simple Notification Service (Amazon SNS)
Amazon Simple Queue Service (Amazon SQS)
App B

Demo 5

Twelve-Factor App:
12: Admin processes

Running admin processes

Admin/ management processes are inevitable:

- Migrate database
- Repair some broken data
- Once a week move database records older than X to cold storage
- Every day email a report to this person

Demo 6

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

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

Earn an industry-recognized credential:
AWS Certified Developer – Associate
AWS Certified DevOps – Professional

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

twitter.com/AWSCloud

facebook.com/AmazonWebServices

youtube.com/user/AmazonWebServices

slideshare.net/AmazonWebServices

twitch.tv/aws

Thank you!

Donnie Prakoso
Sr. Developer Advocate, ASEAN

Amazon Web Services

@donnieprakoso
donnieprakoso
go.donnie.id/youtube
donnieprakoso