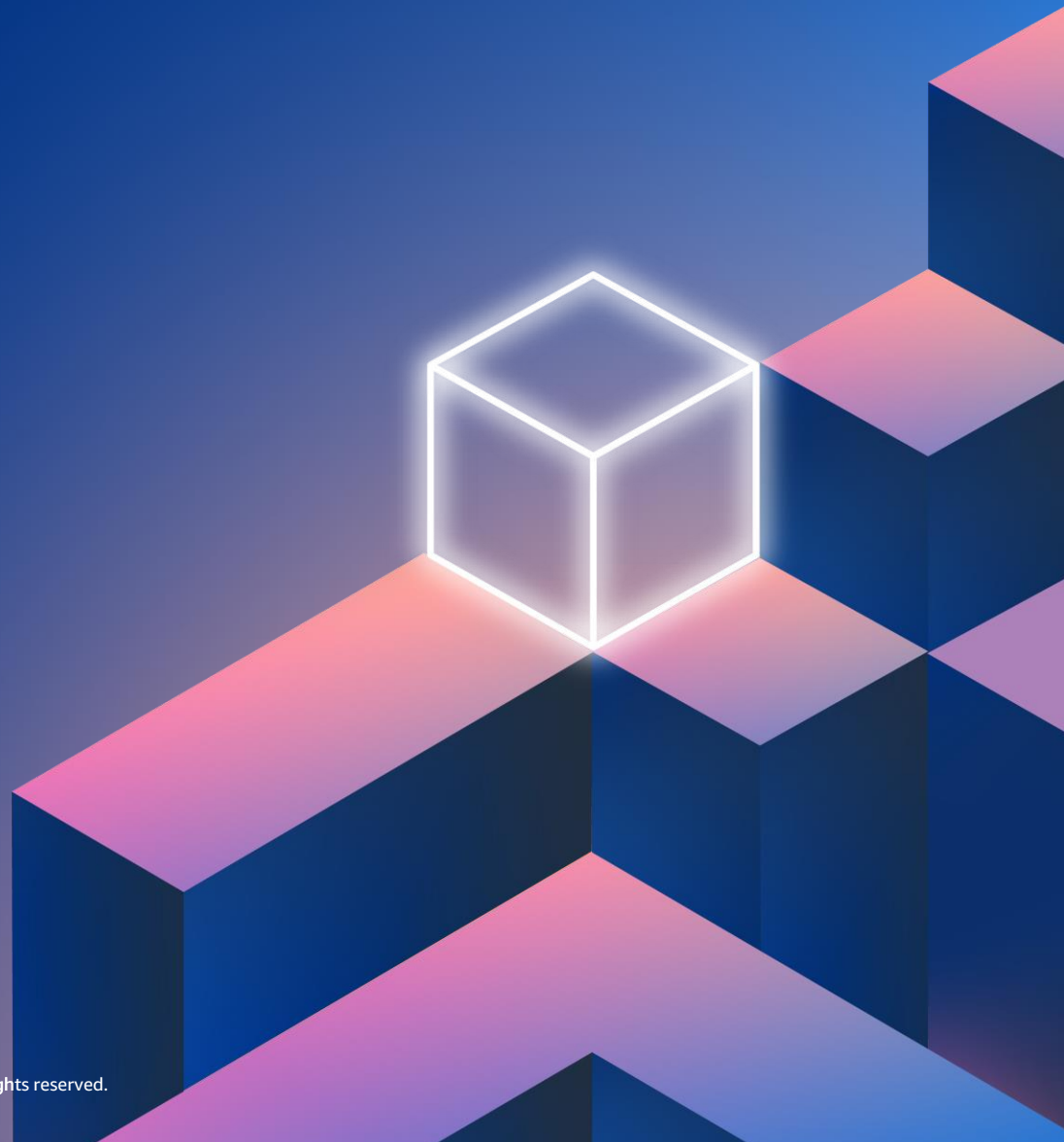
AWS Builders Online Series

Deploy containerized web applications in minutes



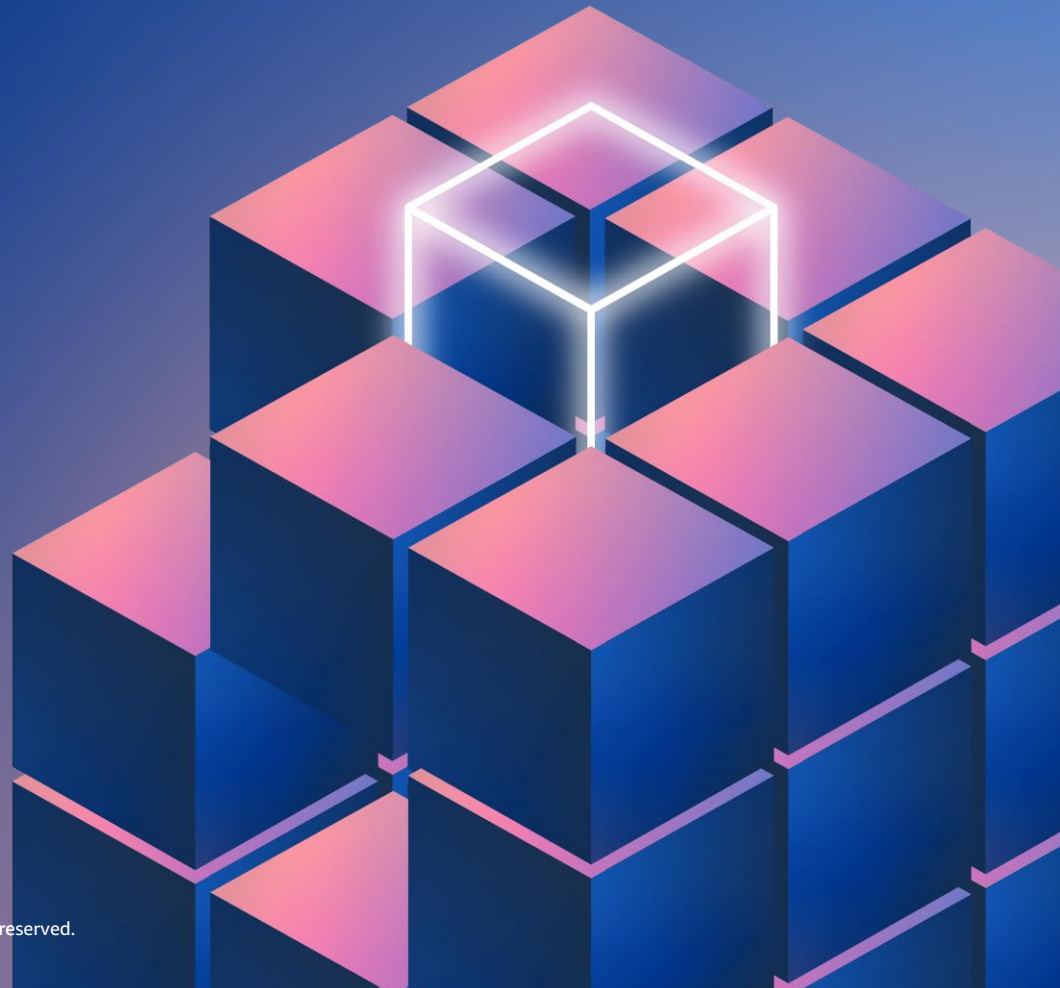
Tuan Huynh

Cloud App Architect
Amazon Web Services

@tuanmhuynh

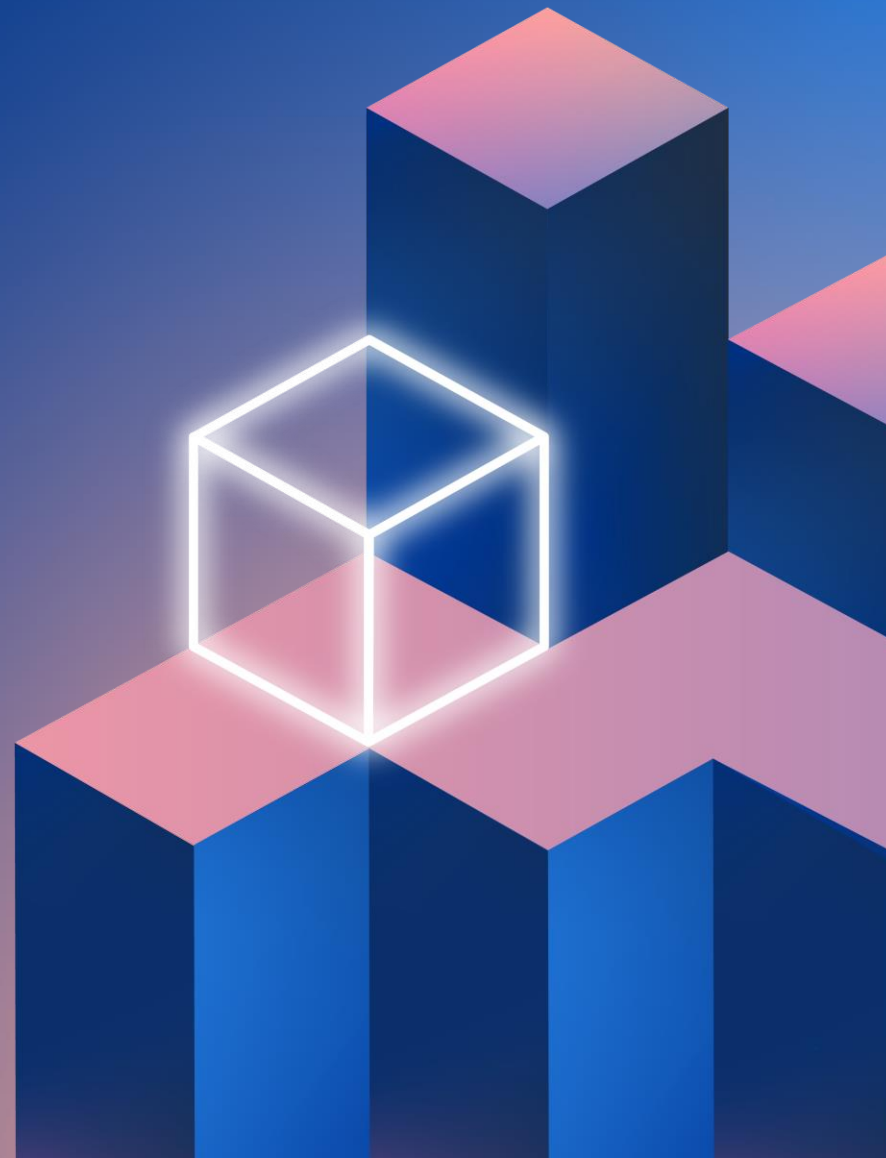
Key takeaways

- Understand the benefits of using containers
- The essential container services on AWS
- Learn how to deploy and scale containerized workloads using AWS App Runner and Copilot



Video 1 - showcase

Why containers?



Applications aren’t just code, they have dependencies

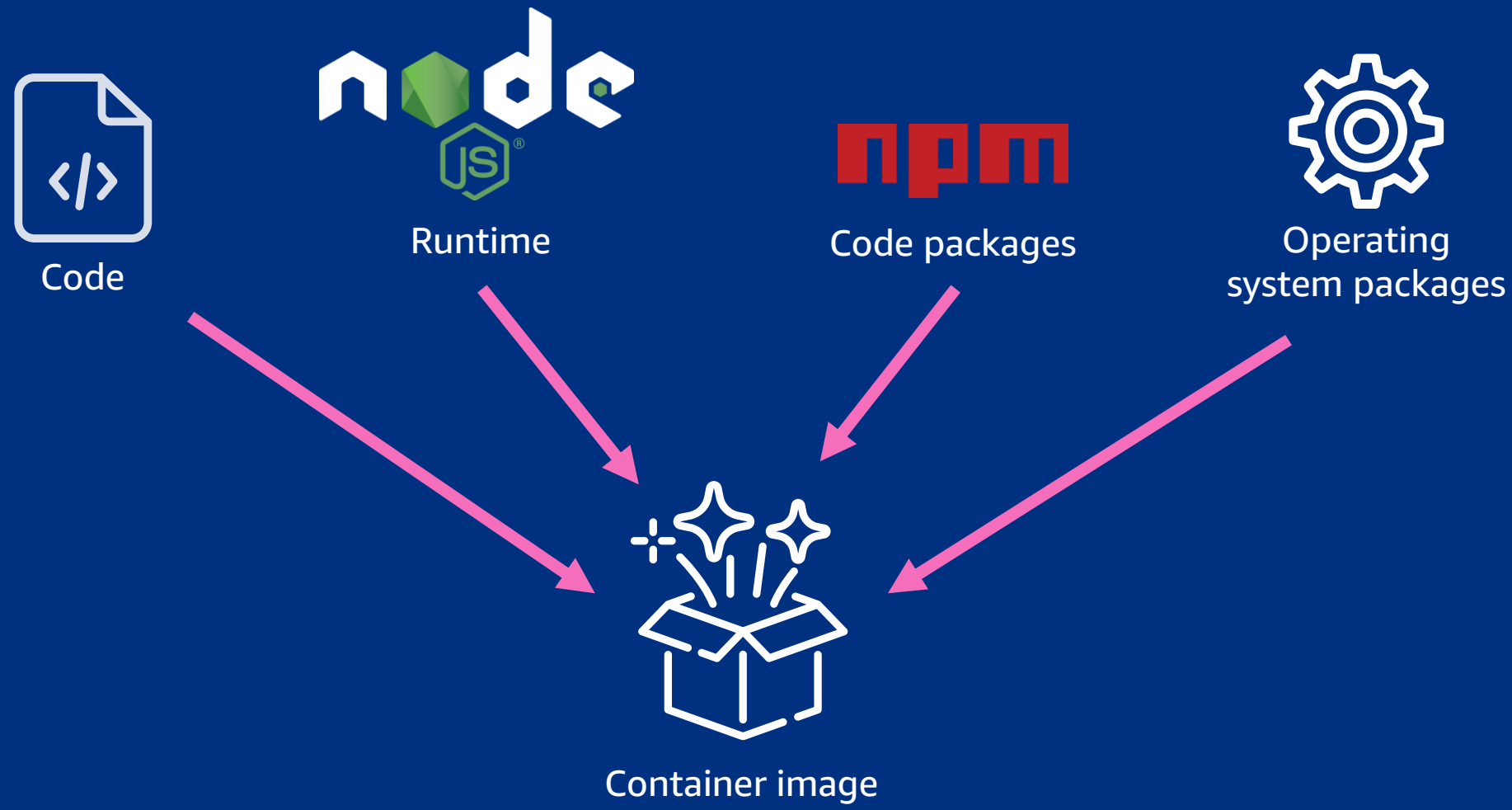
Code

Runtime

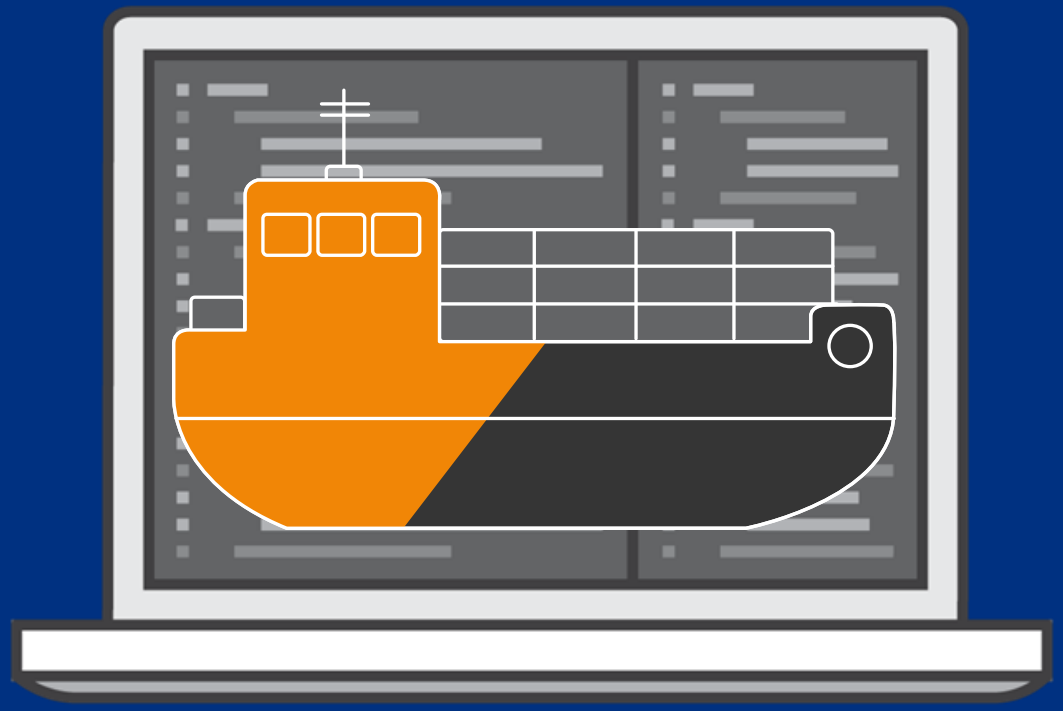
Code packages

Operating system packages

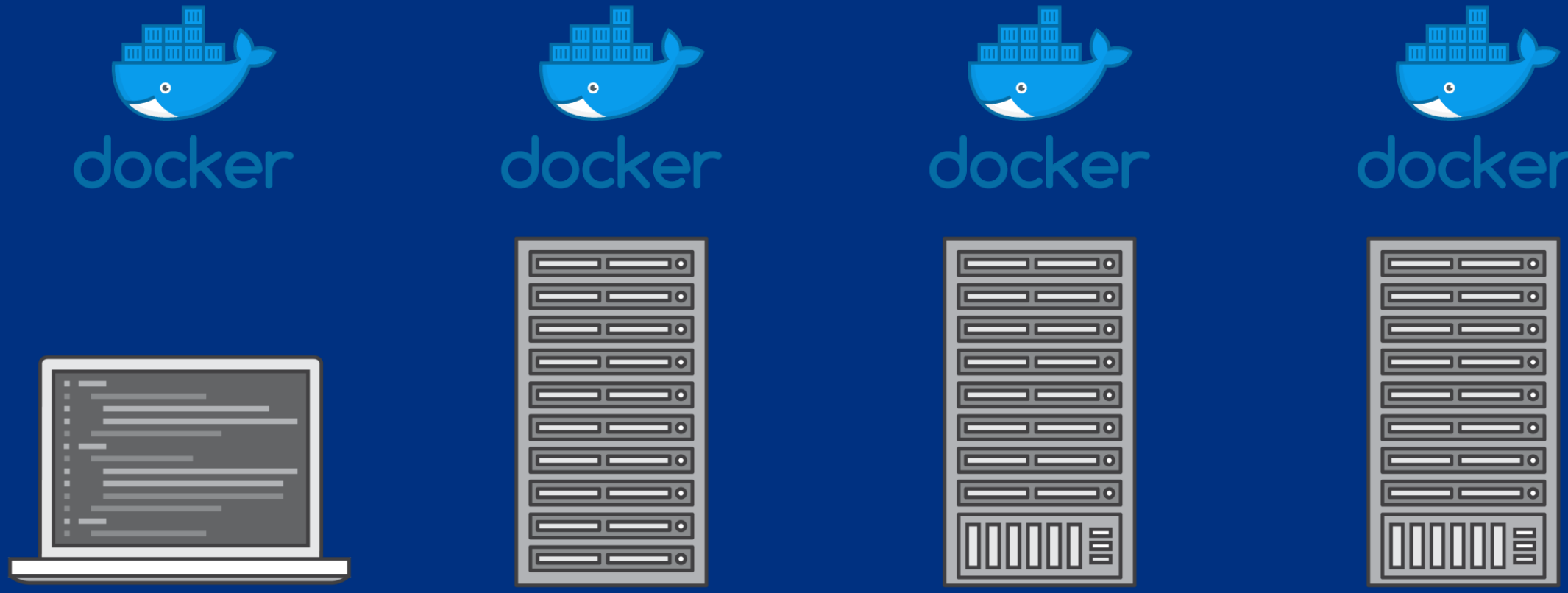
Containers turn applications into one deployable artifact



Running containers in localhost is easy



Four environments, same container



Local laptop

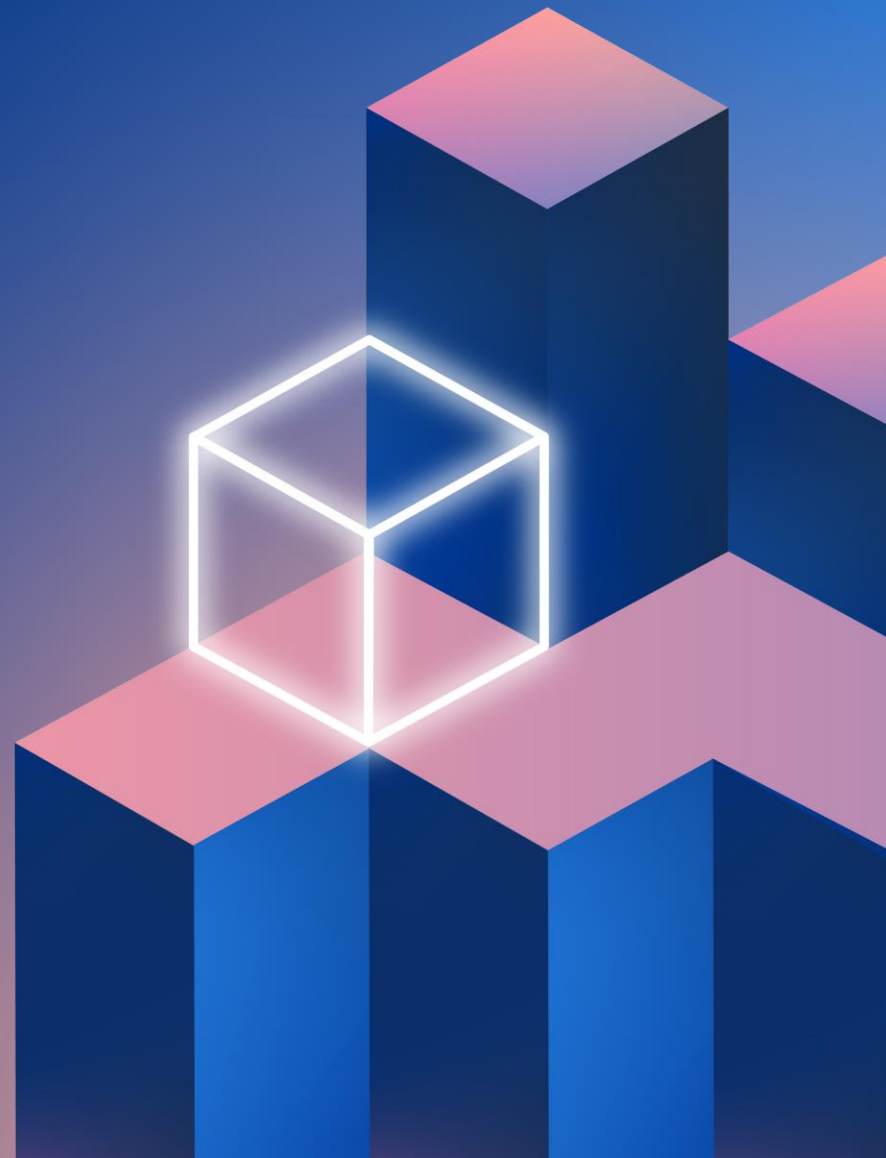
Staging / QA

Production

On-prem

Running containers in production is hard

Container orchestration services



Modern application computing services landscape

COMPUTE

- AWS Lambda
- Amazon ECS
- Amazon EKS
- AWS App Runner

INFRASTRUCTURE

- AWS Fargate
- Amazon EC2

IMAGE REGISTRY

- Amazon ECR

NETWORKING AND INTEGRATION

- Amazon EventBridge
- Amazon SNS
- AWS Step Functions
- AWS AppSync
- Amazon SQS
- Amazon API Gateway
- AWS Cloud Map
- AWS App Mesh

Third-party tooling

DATADOG · sumo logic · sysdig · New Relic. · Prometheus · Grafana · OpenTelemetry

aws

Amazon ECS or Amazon EKS?

Amazon Elastic Container Service (Amazon ECS)

Powerful simplicity

- AWS-opinionated way to run containers at scale
- Reduce decisions without sacrificing scale or features
- Reduce time to build, deploy, and migrate applications
- Great for both short and long running tasks/services

Amazon Elastic Kubernetes Service (Amazon EKS)

Open flexibility

- Gain agility and efficiency with AWS-optimized Kubernetes, and standardize operations everywhere
- Secure, highly available, with observability across all Kubernetes deployments
- Build with choice of solutions from the broader community around Kubernetes
- Great for long running tasks/services, large scale enterprise applications

High performance container registry

Amazon Elastic Container Registry (Amazon ECR)

Deep integration with AWS platform

Integrated with Amazon ECS and Docker CLI

Scalable and highly available

Serverless container

AWS Fargate

Fargate removes the operational overhead of scaling, patching, securing, and managing servers

Improve security through workload isolation by design.

Only pay for what you use. Fargate scales the compute to closely match your specified resource requirements.

Fully managed service for web applications

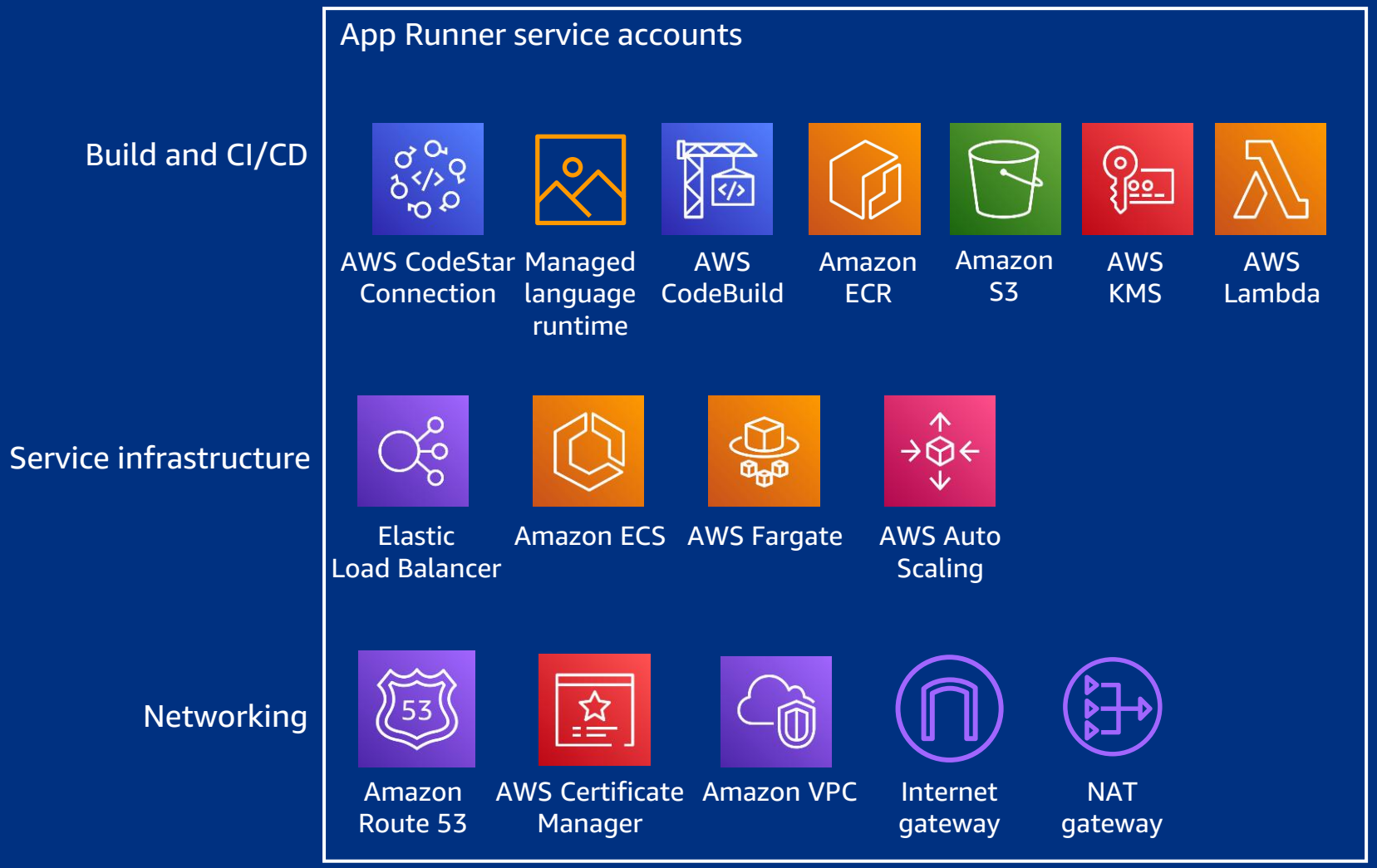
AWS App Runner

AWS App Runner is a fully managed service for web applications.

Build and run secure web applications at scale, without prior container or infrastructure experience.

Scale your applications cost effectively, with high availability and low latency.

AWS App Runner is a simple serverless experience for running HTTP request/reply services
Look at how much we abstract away behind service accounts!



When to use AWS App Runner:

1

Web Applications & API servers

- Applications that serve HTTP based requests

2

Multi-concurrent

- The application is long-running
- A single instance of the application may serve many requests during its lifetime
- Multiple requests maybe handled simultaneously

3

Stateless

- Requests are processed independently and do not depend on local state.
- State maybe stored external to the application instance (eg: a DynamoDB table)

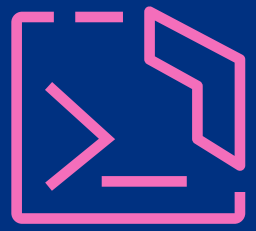
4

No background processing

- Any processing outside the context of a request must be limited


aws

Simplicity + serverless =

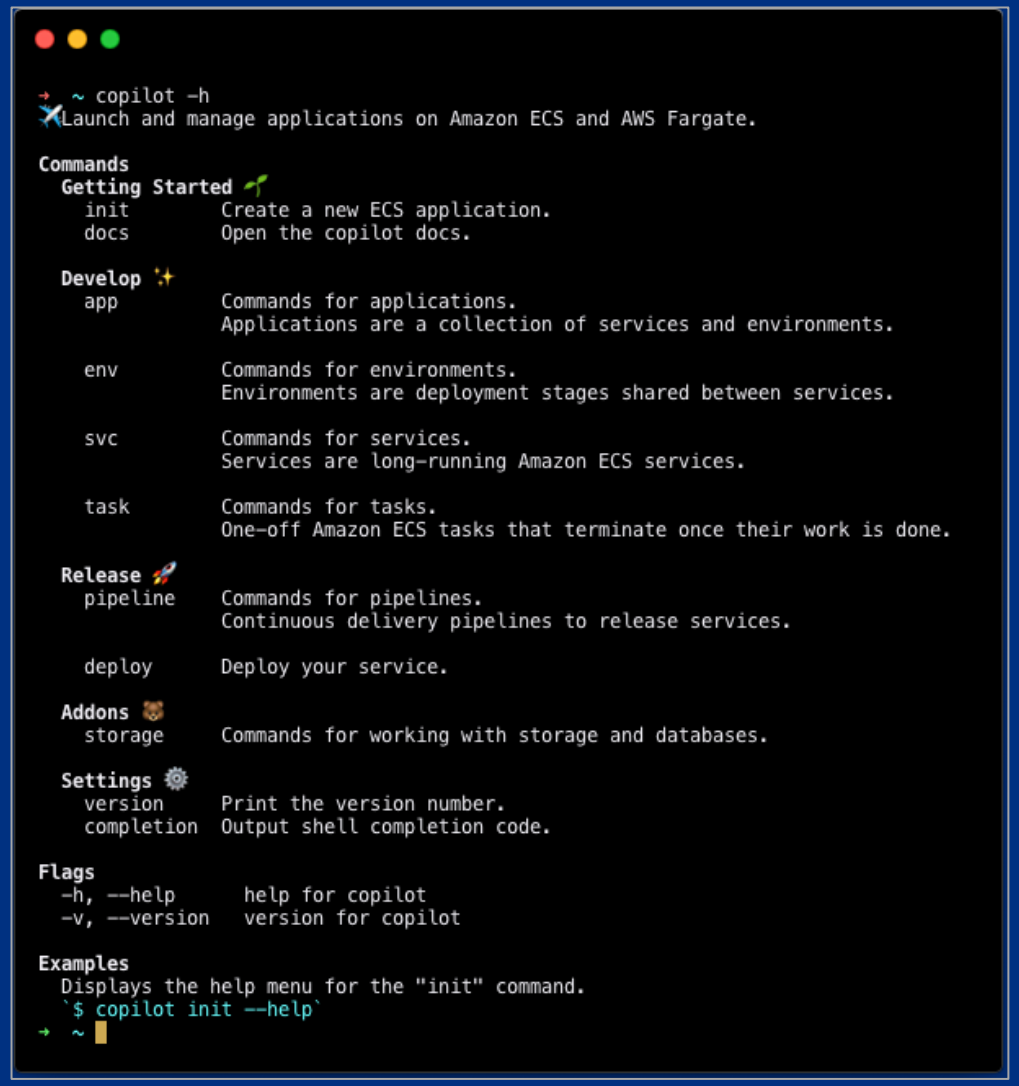


AWS
Copilot

Amazon ECR

AWS App Runner

Introducing AWS Copilot CLI



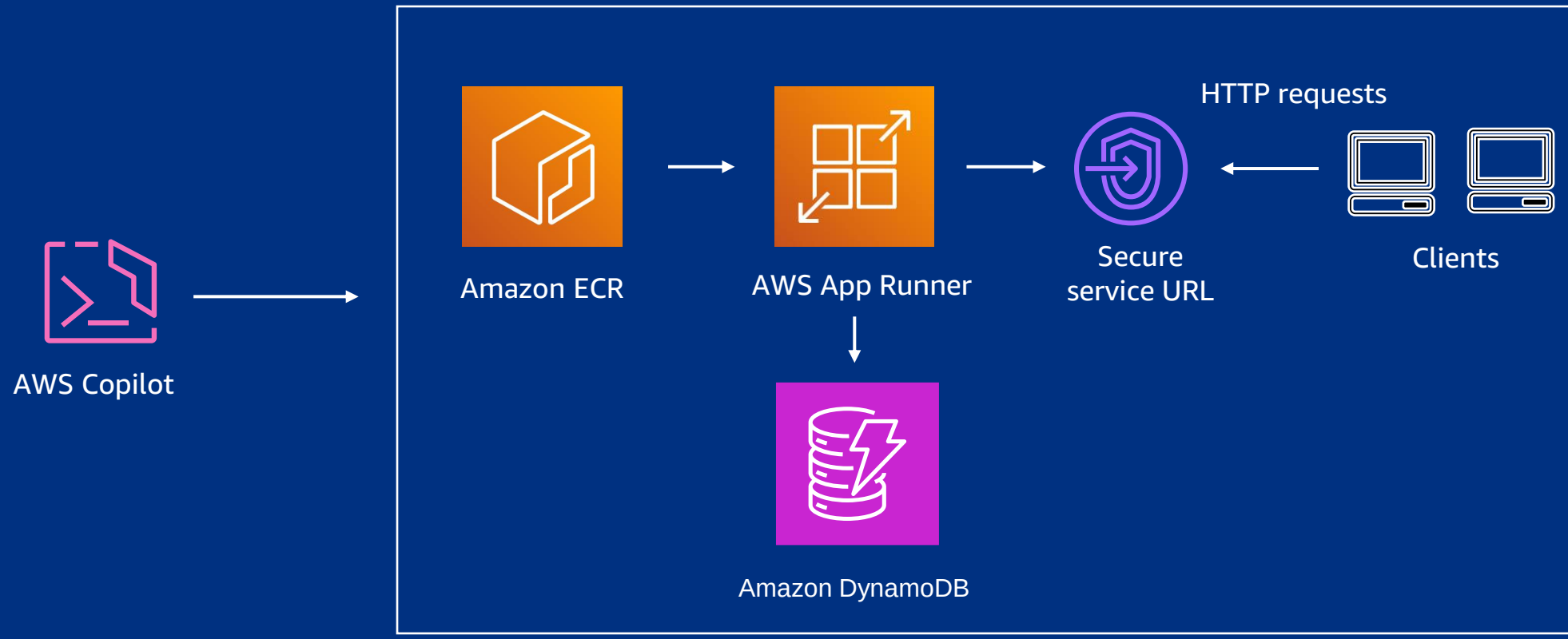
Solve all of these challenges

- 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?
- ...and more!

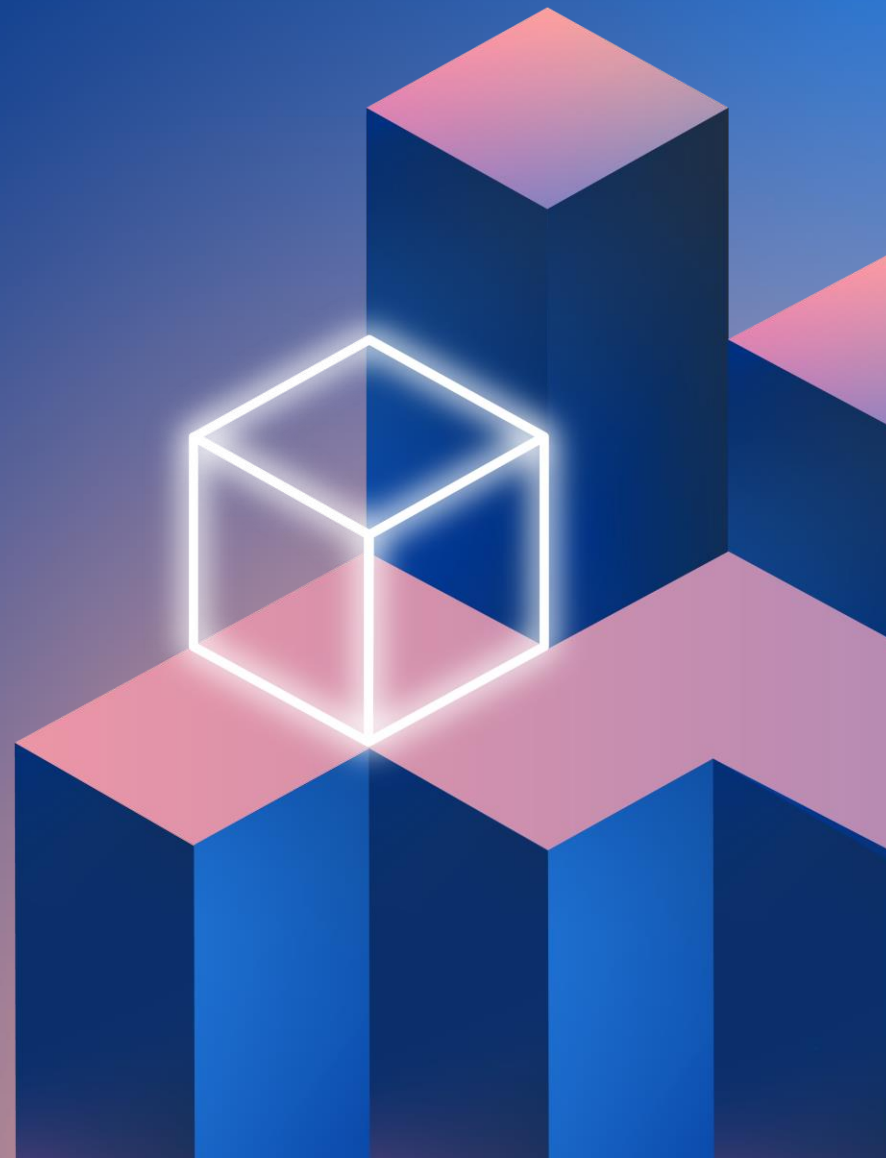
How to deploy applications?

```
$ copilot init
$ copilot svc deploy
```

Flexible application builds and deployments



Demo



How about scaling?

- **Max Concurrency**
- The maximum number of simultaneous requests a single application instance can handle

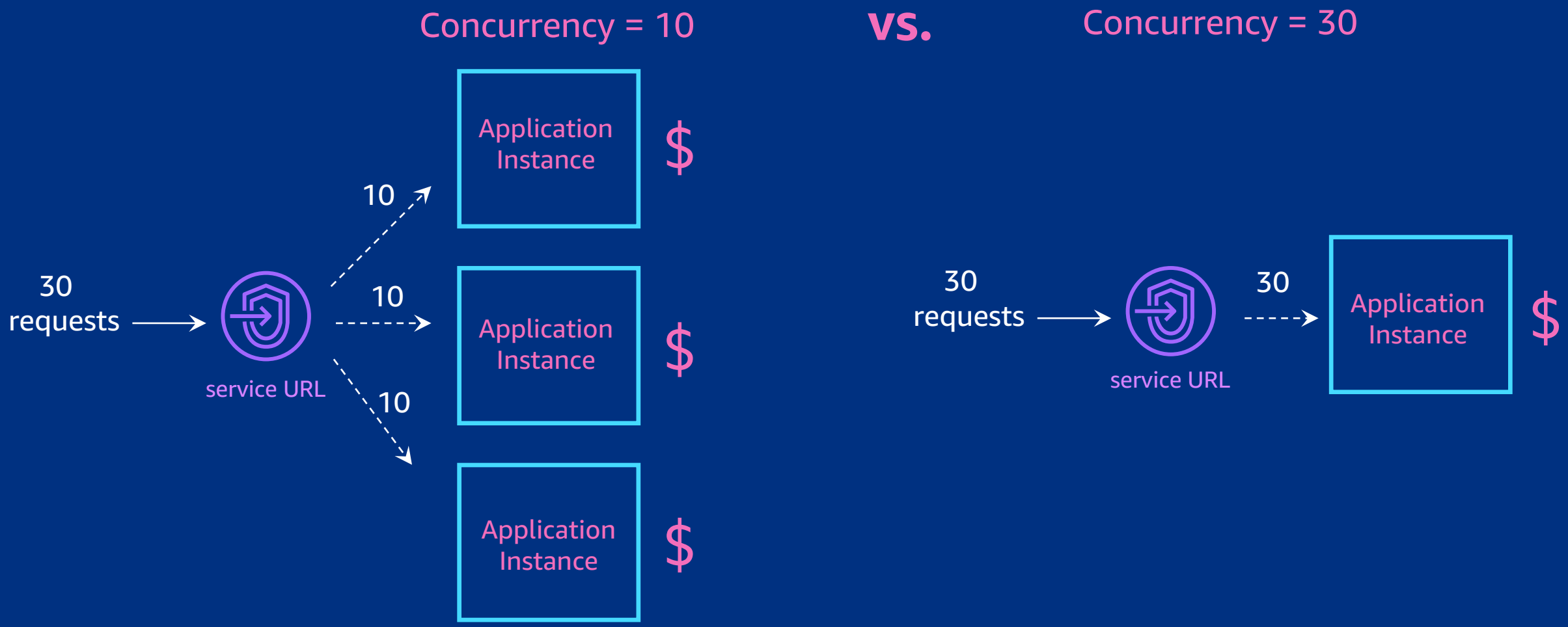
- **Minimum Provisioned Instances**
- Minimum provisioned instances to avoid cold start latencies. Minimum is 1.

- **Maximum Instances**
- Upper bound on the number of instances launched to control cost

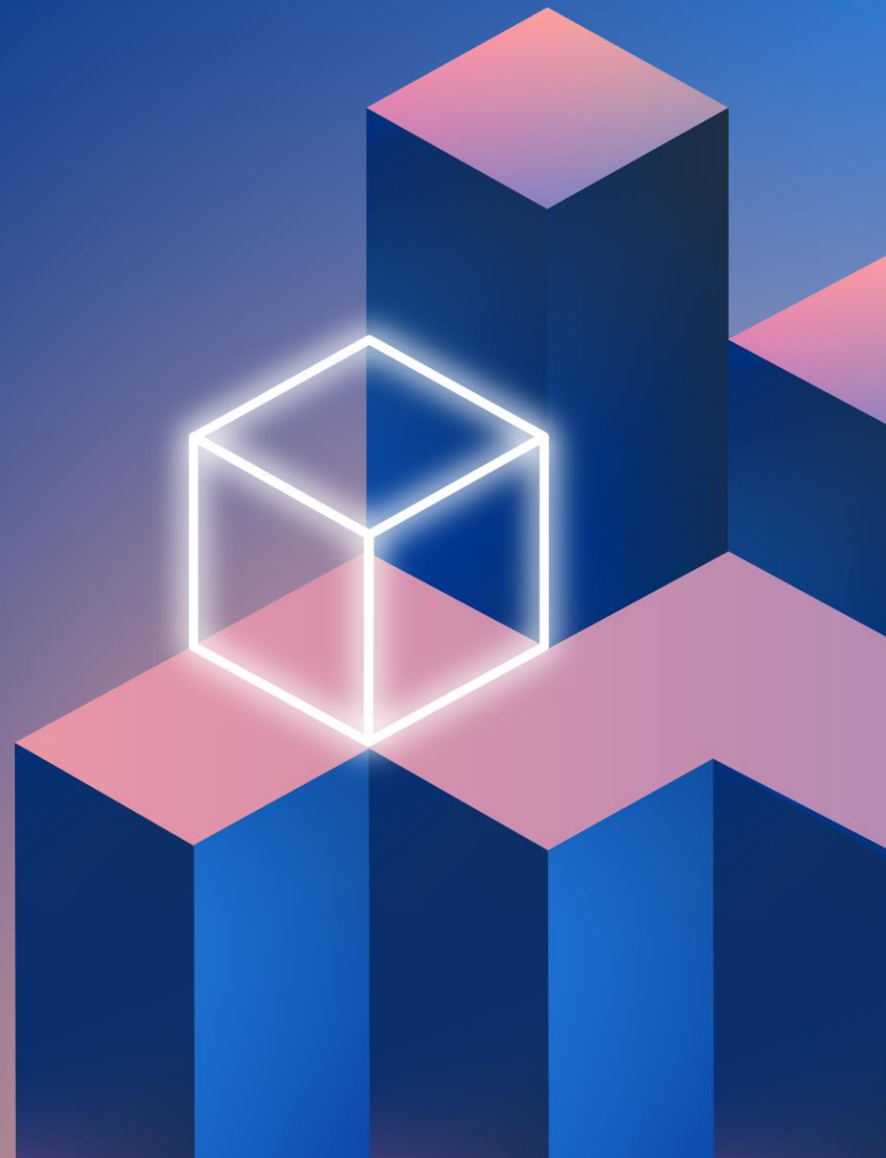
---

- **Pause/Resume Service**
- Services can also be temporarily disabled using the Pause/Resume feature
- While paused, App Runner reduces application instances to zero until you resume

Setting concurrency

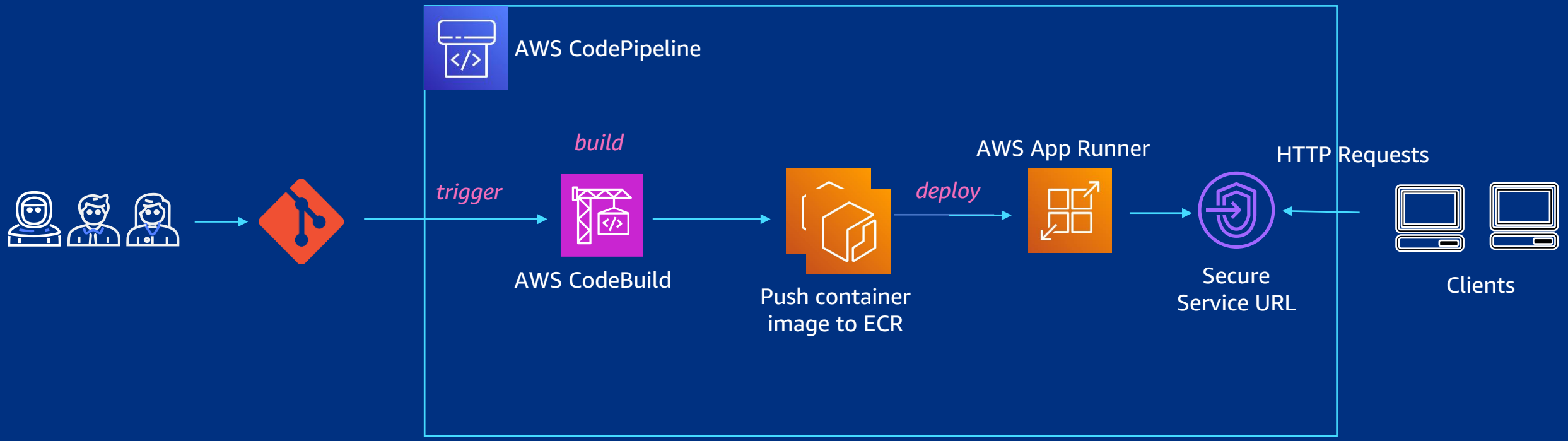


Demo



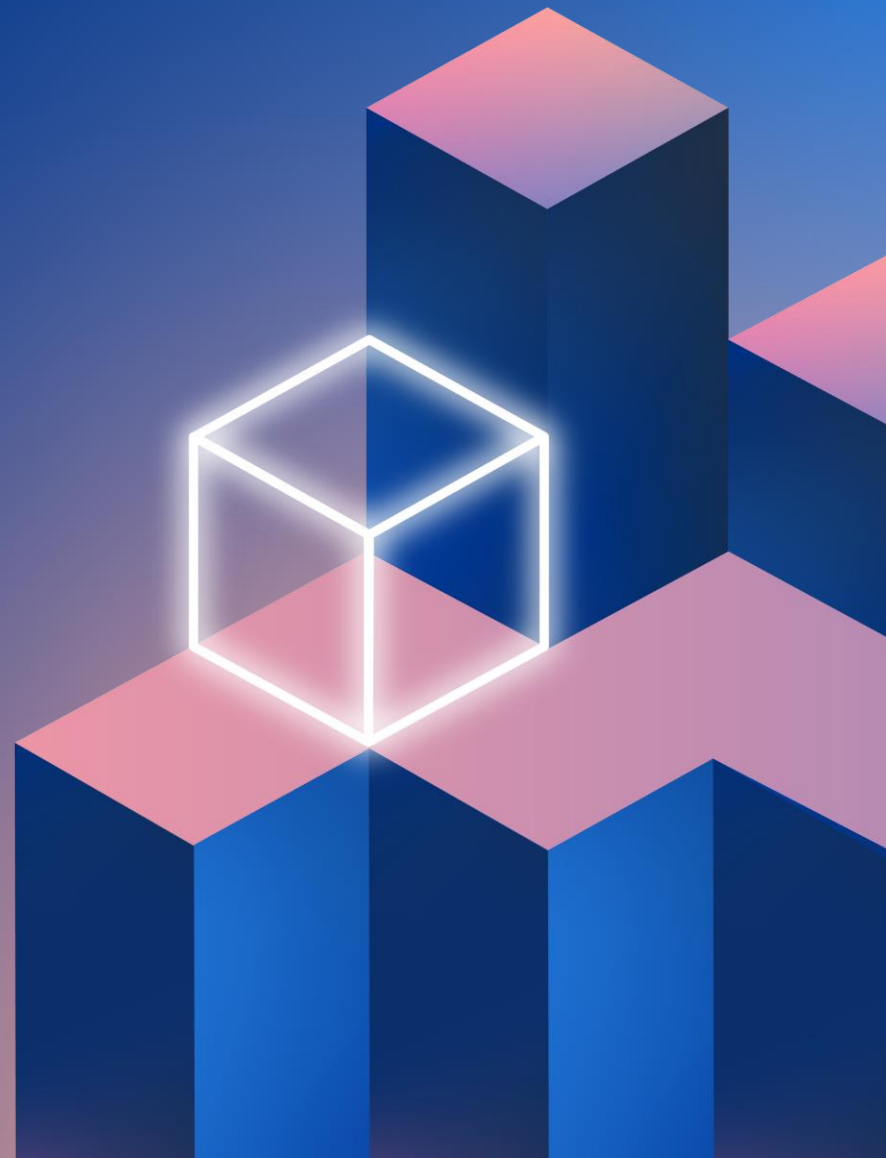
How to implement CI/CD? `$ copilot pipeline init`

Building release pipeline



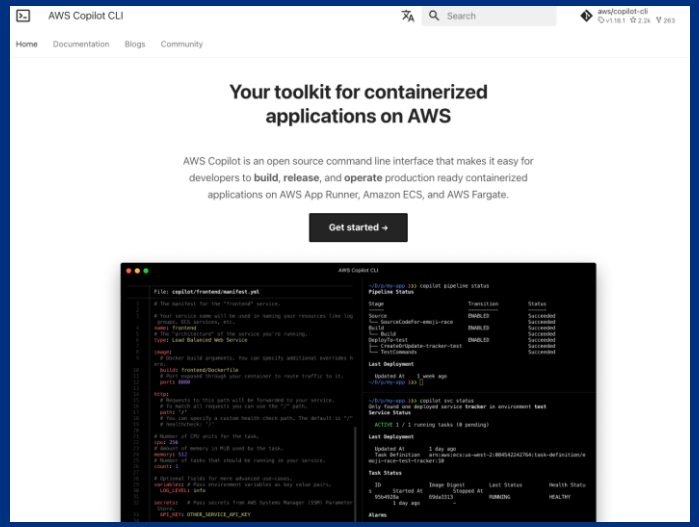
AWS Copilot

Demo

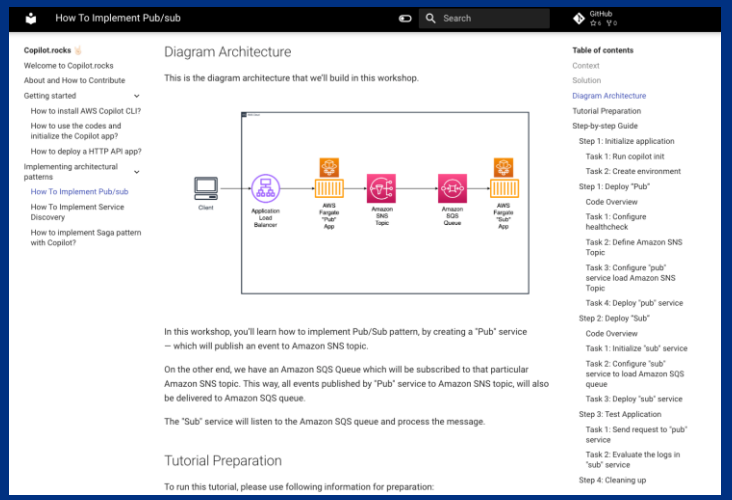


Additional resources

AWS Copilot product page —
https://aws.github.io/copilot-cli/



Copilot.rocks 🤘 —
https://www.copilot.rocks/



AWS App Runner Workshop 🚀 —
https://www.apprunnerworkshop.com/

AWS Training & Certification

Access 600+ free digital courses with AWS Skill Builder

Focus on the cloud skills and services that are most relevant to you across 30+ AWS solutions, including digital self-paced learning plans and ramp-up guides.

LEARN YOUR WAY, EXPLORE SKILLBUILDER.AWS »

Validate your cloud expertise with an AWS Certification

Take the step towards earning an industry-recognised credential. Learn more about how to become an AWS Certified Cloud Practitioner, and AWS resources that can help you prepare.

ACCESS RESOURCES TO PREPARE FOR YOUR EXAM »

aws

Thank you for attending AWS Builders Online Series

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


linkedin.com/company/amazon-web-services


twitch.tv/aws

Thank you!

Tuan Huynh

Cloud App Architect
Amazon Web Services

@tuanmhuynh