



Builders Online Series

AWS 에서 DevOps 시작하기

정영준

AWS 솔루션즈 아키텍트

Table of contents

- DevOps 소개
- 아마존의 DevOps 접근 방법
 - Culture
 - Practices
 - Tooling
- DevOps 를 위한 AWS 도구들 알아보기
 - 운영 책임 모델의 변화
 - 소프트웨어 딜리버리의 발전
- 추가정보

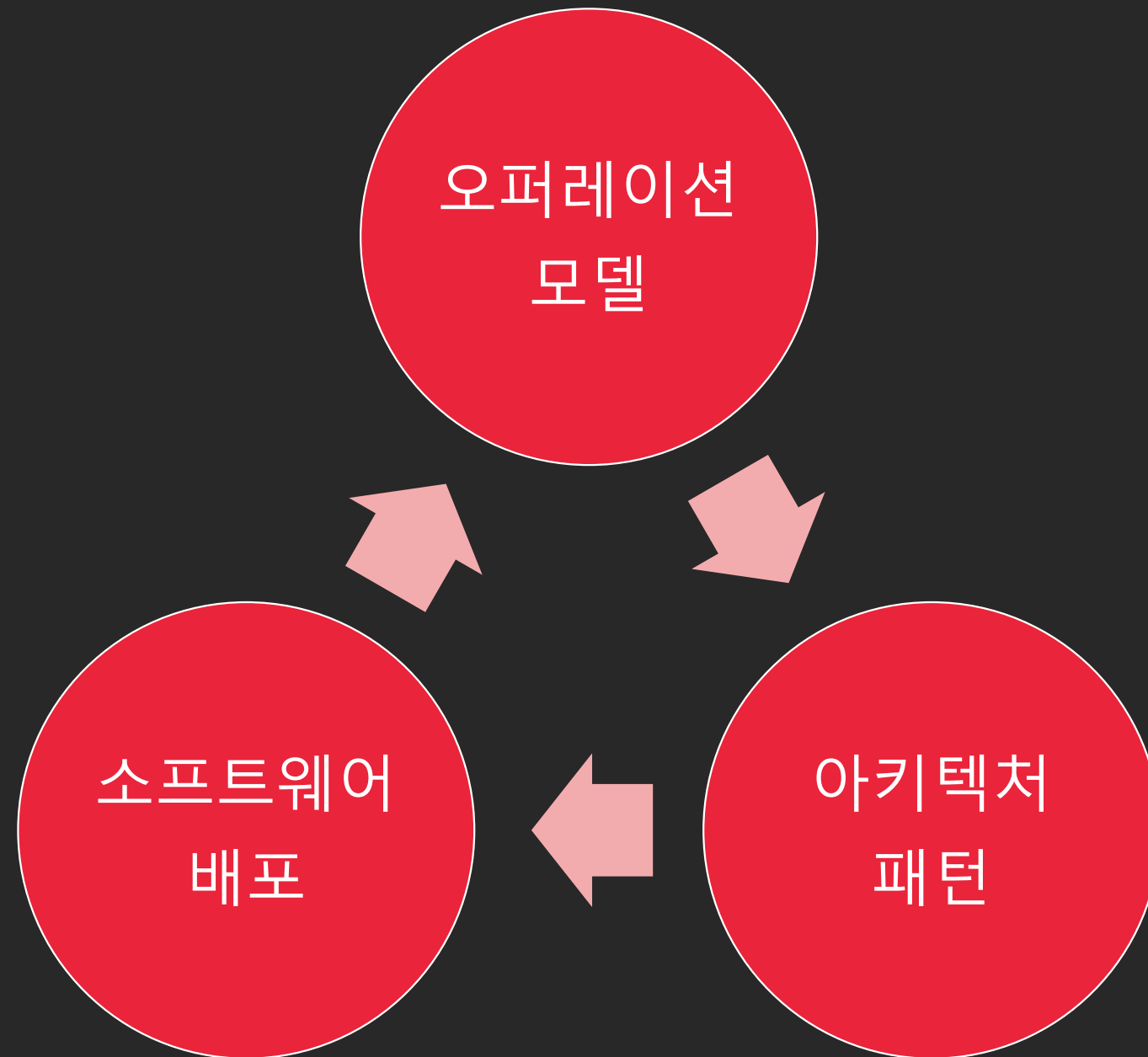
DevOps 소개



DevOps 란 무엇일까요?

소프트웨어 개발과 정보 기술 운영을 결합한
문화 철학, 사례 및 도구의 조합입니다.
이러한 사례를 통해 기업은 새 애플리케이션
기능과 개선된 서비스를 더욱 빠른 속도로
고객에게 전달할 수 있습니다.

혁신적인 DevOps 를 위한 변화



Amazon 의 DevOps



유연함을 위하여 작은 단위로 나눔
(마이크로서비스, 2 피자팀)



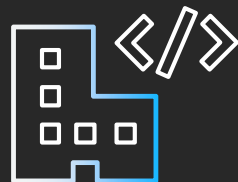
모든 것을 자동화



표준 도구를 사용



프로세스 공정화
(프로세스 표준화, 템플릿)

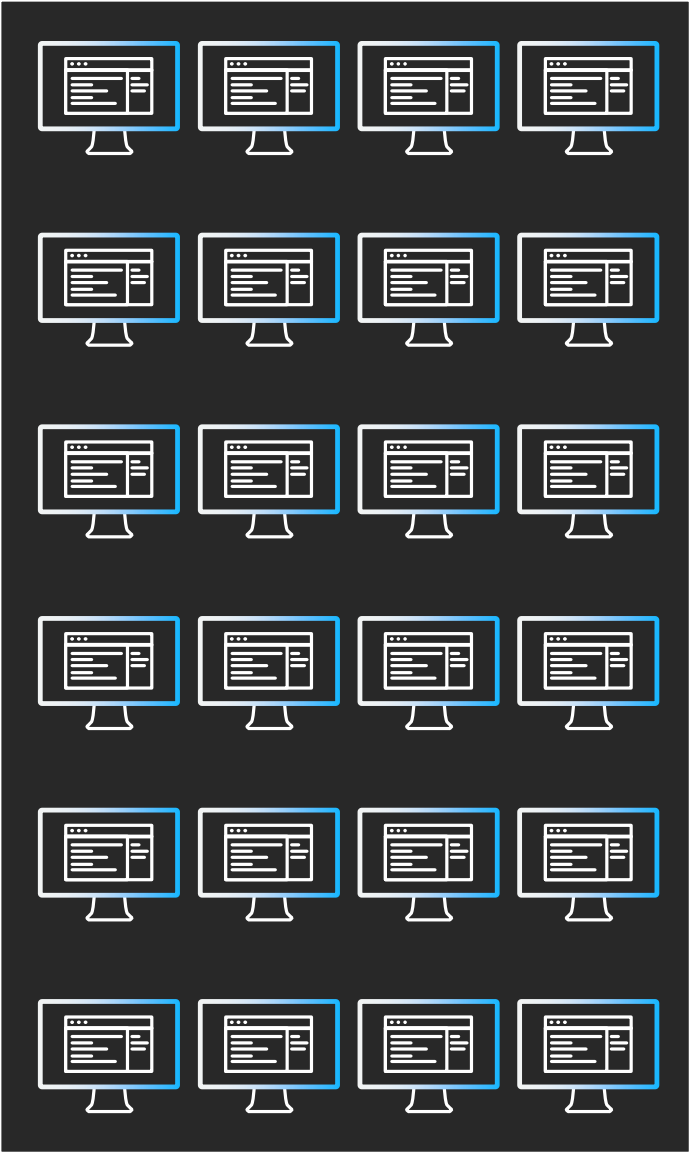


코드로서의 인프라

모놀리스 개발 라이프 사이클

개발자

서비스

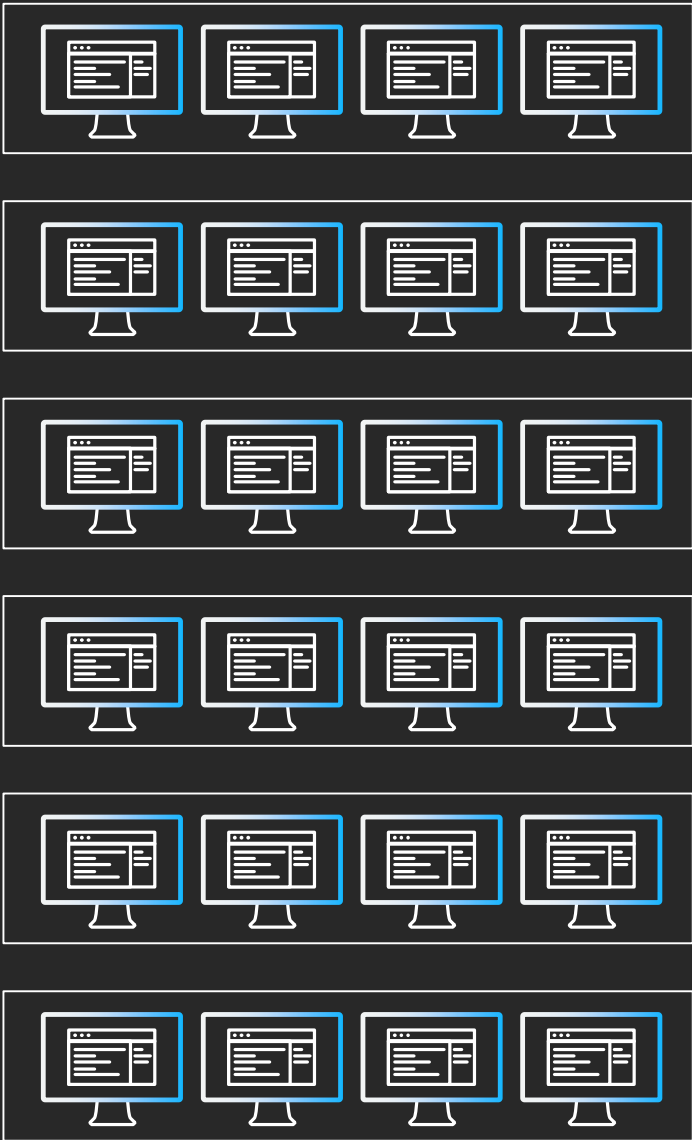


운영 파이프 라인

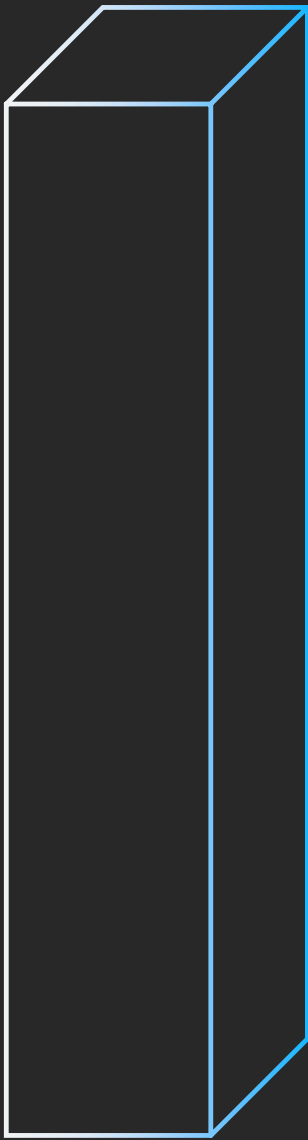


모놀리스 개발 라이프 사이클

개발자



서비스



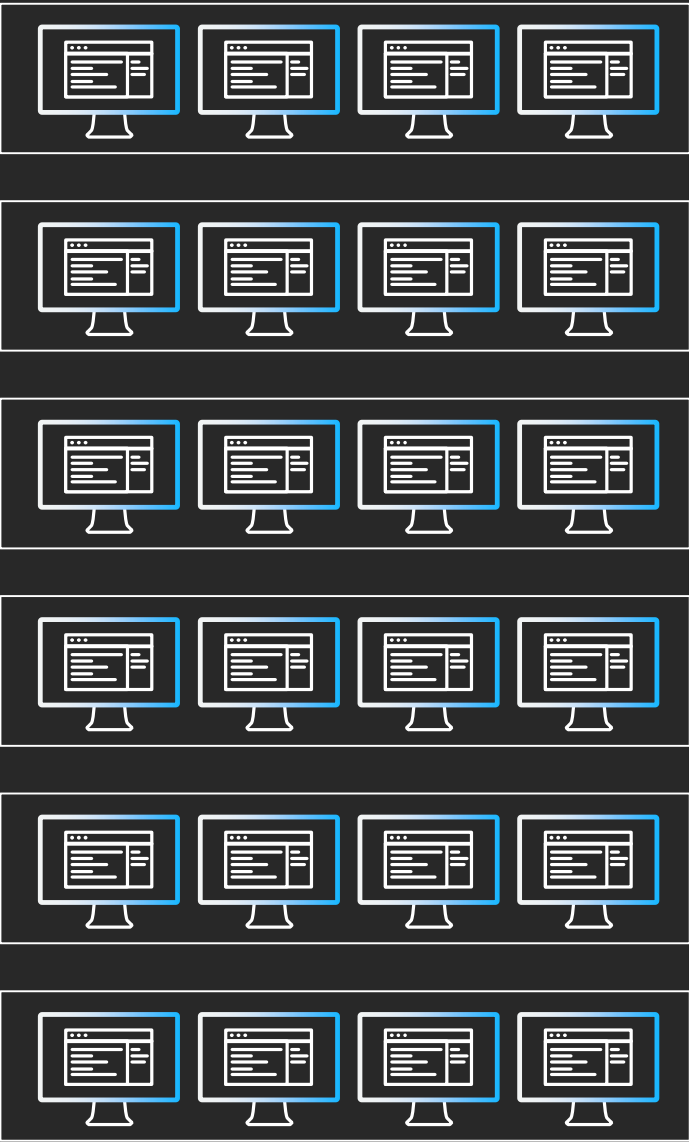
운영 파이프 라인



모놀리스 개발 라이프 사이클

개발자

서비스



운영 파이프라인

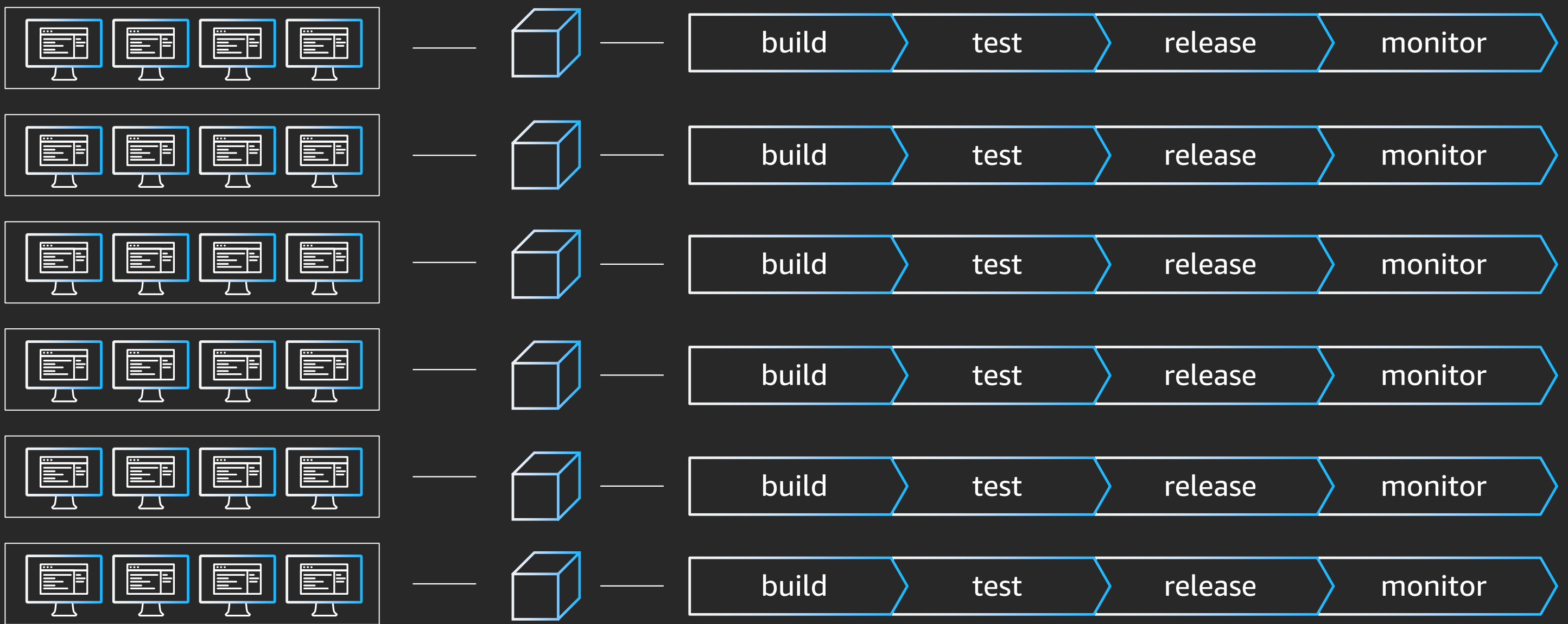


Microservice development lifecycle

개발자

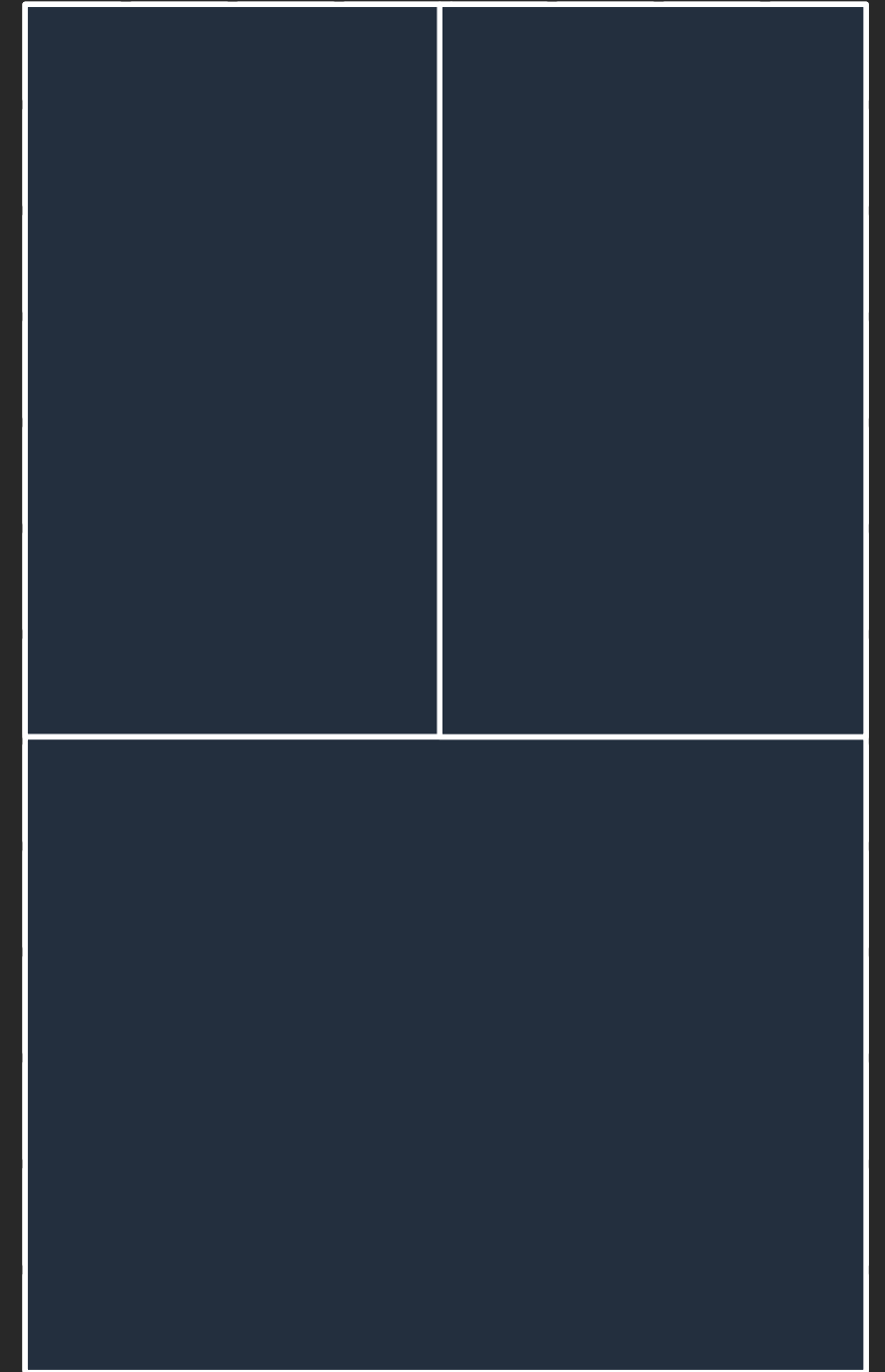
서비스

운영 파이프 라인

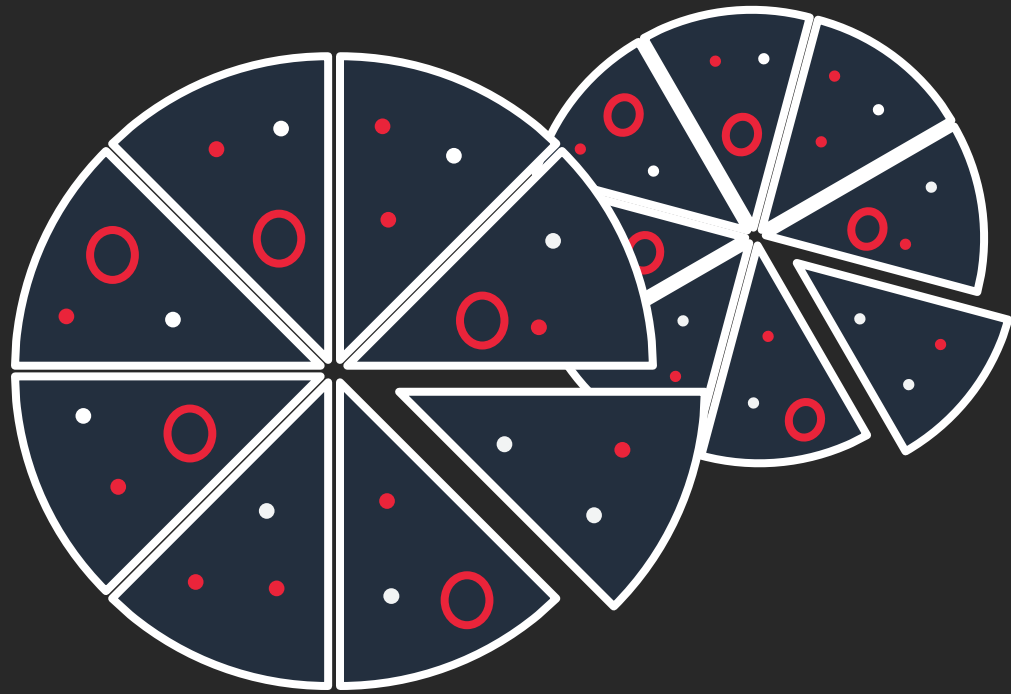


작은 단위로 서비스를 나누는 원칙

- 유닛을 가능한 한 작게 (Primitives)
- 데이터 도메인 생성
- 스케일링(확장) 단위 기반으로 분리 (기능단위 X)
- 각각의 서비스를 별도의 운영
“Communication is terrible!” —Jeff Bezos
- APIs (contracts)를 사용한 서비스간 통신



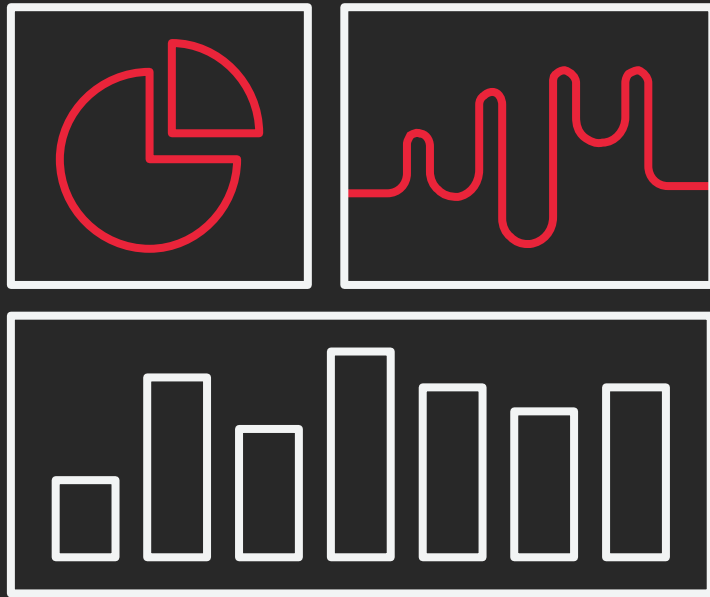
조직 (재)구성



“투-피자” 팀

- 서비스를 직접 운영
- 사회적 제약 최소화 (Conway의 법칙)
- 결정을 내리는 자율성

조직 (재)구성



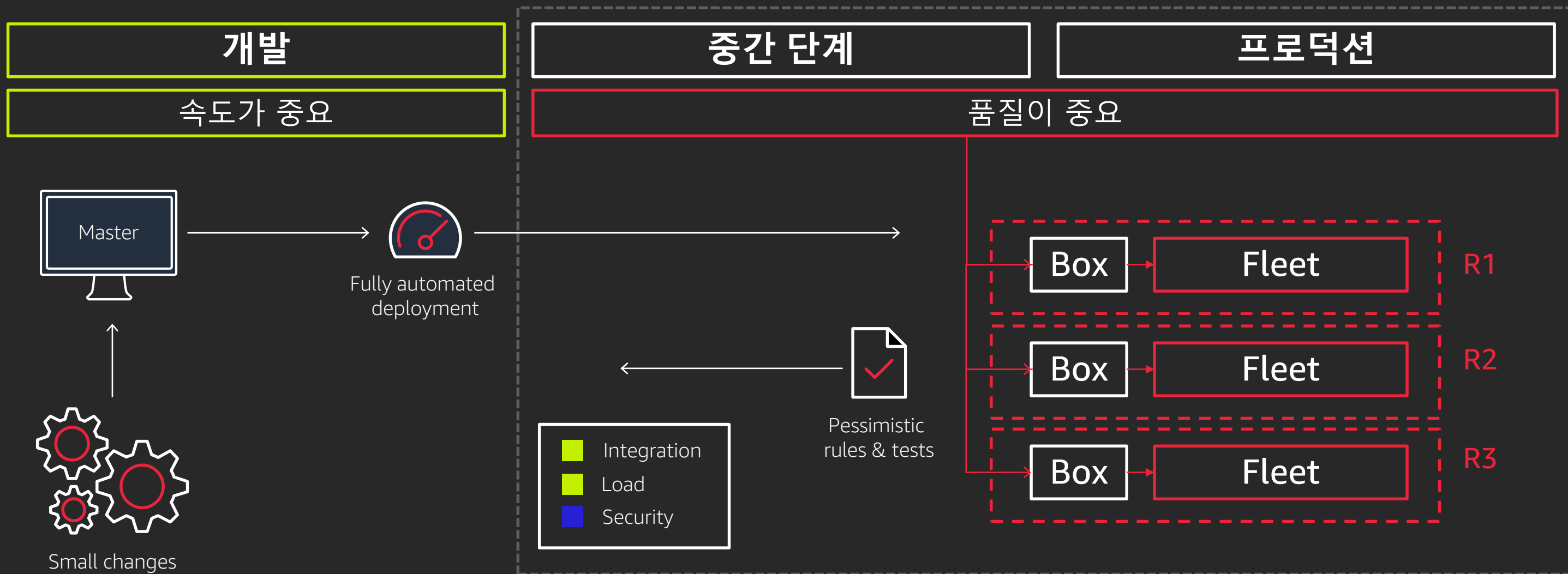
모든 것을 개발팀이 직접 한다

- 계획
- 보안
- 성능
- 확장
- 배포/전개
- 운영
- 버그
- 문서화
- 테스트...

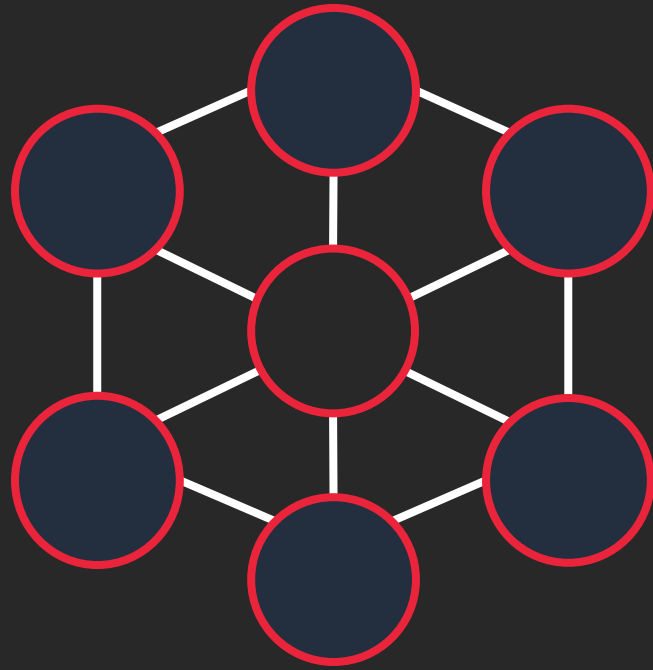


작은 개발팀이 어떻게 이것을 다 할 수 있을까요?

모든 것을 자동화



현대적 애플리케이션 요구사항



현대적 애플리케이션

- 한가지 목적
- 독립 배포
- 독립적 인 확장
- 변화에 따른 영향력을 최소화
- 개별적인 기술 선택

현대적 애플리케이션 요구사항



현대적 애플리케이션

- 독립적으로 확장 가능한
마이크로 서비스 사용
(서버리스, 컨테이너 ...)
- API를 통해 연결
- 지속적으로 업데이트 제공
- 변화에 빠르게 적응
- 글로벌 수준으로 손쉽게
확장
- 내결함성
- 상태 및 지속성을주의
깊게 관리
- 보안 기능 내장

다양한 산업군에 적용되는 DevOps

OSCAR

2 명의 시스템
엔지니어
45 명 이상의 개발자
셀프 서비스
인프라 도구
HIPAA 요구 사항



1000 개 이상의
인스턴스를 5 명의
운영 인력

GILT

수백 개의 마이크로
서비스
AWS를 확장하는
셀프 서비스 도구
거의 전적으로 t2
인스턴스



"성수기 동안 40 개
이상의 동시 실험
배포"



마이너 리그 게임 및
이벤트 중에
대규모로
확장하였다가
비시즌에는 섯다운



DevOps를 위한 AWS 도구

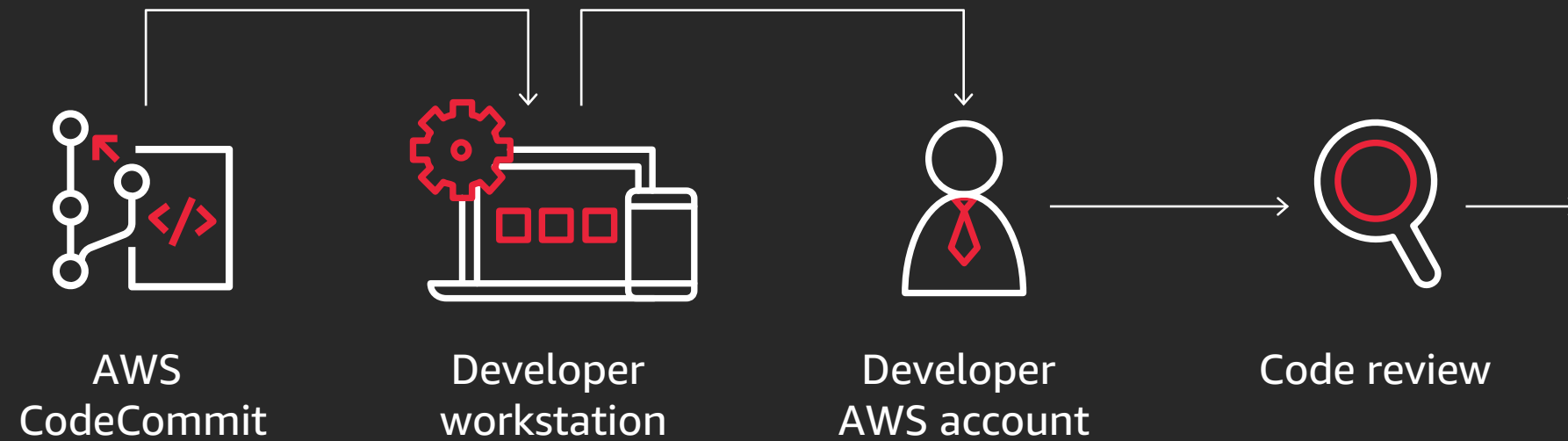


DevOps 파이프라인 예제

개발단계

중간단계

프로덕션단계



- 가능한 한 가장 범위로 작업
- 지역 코드 분기 만
- 로컬에서 개발 및 테스트
- 종속성을 명시 적으로 정의 (테스트에 필요한 종속성 포함)
- 검토를 위해 변경 사항 게시

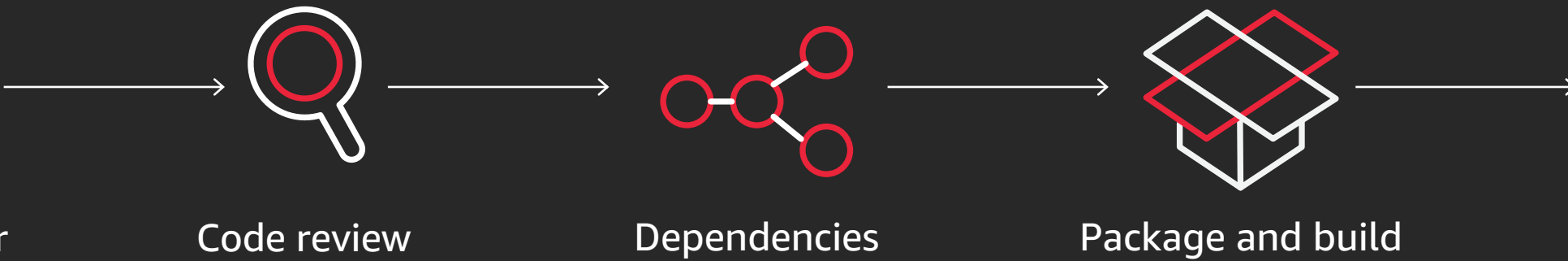
개발과 배포, 테스트에서 빠른 **속도가 중요**

DevOps 파이프라인 예제

개발단계

중간단계

프로덕션



- 단위 테스트 빌드 및 실행
- 코드 및 런타임 종속성을 결합된 아티팩트로 번들링
- 의존성 상관관계 추적

철저한 검증이 중요

DevOps 파이프라인 예제

개발단계

중간단계

프로덕션

성공시에만 승격, 테스트 실패 롤백, 베이킹 실패시 중지



dependencies



Package and build

베타

- 환경만 분리하여 가혹한 테스트
- 파괴적 또는 기타 테스트 실행

감마

- 격리되었지만 "프로덕션"과 같은 환경에서 테스트
- 완전한 '엔드 투 엔드' 통합 테스트 실행

프로덕션

- 작게 시작하여 팬아웃
- 웨이브로 배포하고 각 웨이브 시간을 '베이킹'할 시간을 줍니다.
- 베이킹 모니터링은 일반적으로 더 자세하게 상황 모니터링

Looking for any reason to **fail**

CI/CD 를 위한 AWS 개발자 도구



AWS CodePipeline

소스

빌드

테스트

배포

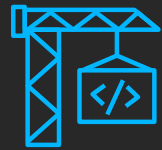
모니터



AWS CodeCommit



AWS CodeBuild



**AWS CodeBuild +
Third Party**



AWS CodeDeploy



AWS X-Ray



Amazon CloudWatch

AWS 의 다양한 엔터프라이즈급 DevOps 도구

CI/CD Tools



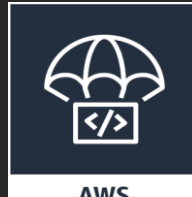
AWS CodeStar



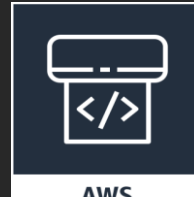
AWS CodeBuild



AWS CodeCommit

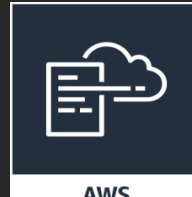


AWS CodeDeploy

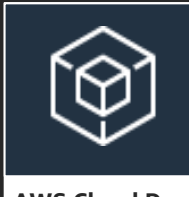


AWS CodePipeline

Infrastructure as Code

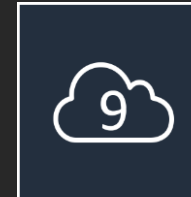


AWS CloudFormation



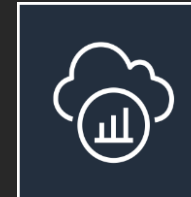
AWS Cloud Dev. Kit (CDK)

IDE

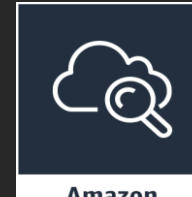


AWS Cloud9

Monitoring & Tracing



AWS X-Ray



Amazon CloudWatch

Web Apps



AWS Elastic Beanstalk

IDE and DevOps Toolkits



Visual Studio Code



IntelliJ



PyCharm



Visual Studio



Eclipse



VSTS

CLI and Scripting Tools



AWS CLI



Tools for PowerShell

Languages



Amazon Corretto

Mobile



AWS Amplify

SDKs



JavaScript



Python



PHP



Ruby



.NET



Java



Go



Node.js



C++

Machine Learning



Amazon CodeGuru



Amazon DevOps Guru

AWS 개발자 도구는 서버리스 마이크로 서비스를 지원하도록 특별히 제작되었습니다.

2016

2017

2018

2019

2020

NOV

NOV

DEC

OCT

NOV

NOV

DEC



- Support for Lambda deployment with CodePipeline and CloudFormation

- Support for rolling and blue/green Lambda deployments with CodeDeploy

- Support for Fargate and ECS deployments in CodePipeline

- CodePipeline supports Config for improved governance

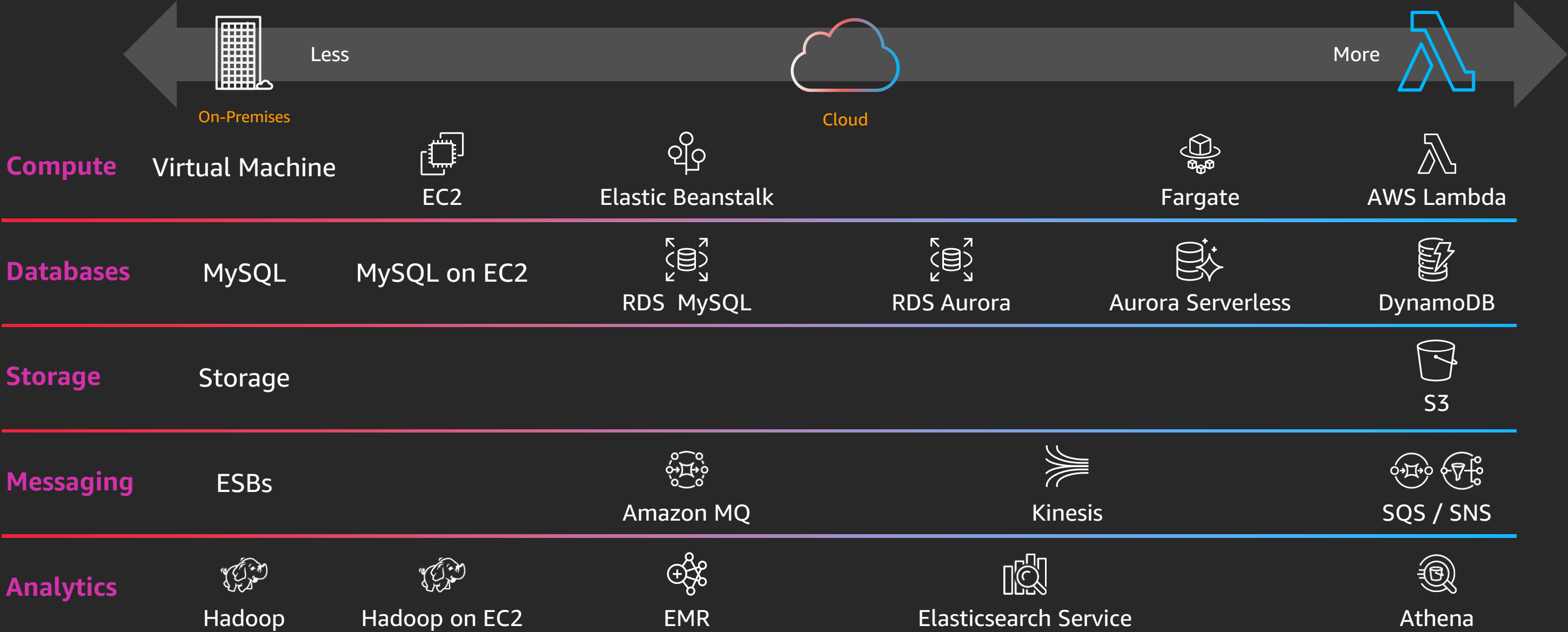
- CodePipeline supports ECR as a source
- Support for blue/green deployments for Fargate and ECS with CodeDeploy
- Support for serverless python apps in AWS Toolkit for PyCharm

- General availability of the AWS CDK
- General availability of AWS Toolkit for IntelliJ and AWS Toolkit for VS Code w/support for serverless apps

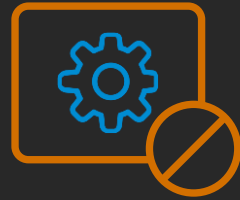
- General availability of the DevOps Guru
- General availability of Code Guru Python Support

DevOps 에서의 운영 책임 모델 변화

AWS 운영 책임 모델



서버리스란 무엇입니까?

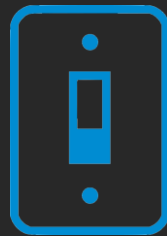


인프라 프로비저닝,
관리가 필요 없음



자동화된 확장

사용한 만큼 지불



고가용성 및 보안

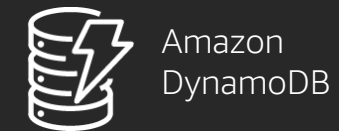
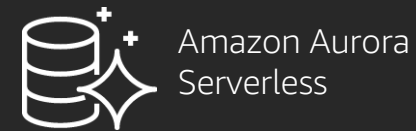


AWS 서버리스 서비스

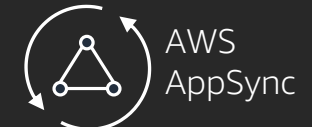
COMPUTE



DATA STORES



INTEGRATION





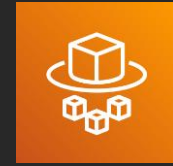
Amazon Elastic
Container Kubernetes



Amazon Elastic
Container Registry



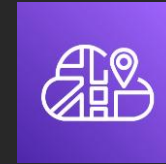
Amazon Elastic
Container Service



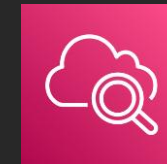
AWS Fargate



AWS App Mesh



AWS Cloud Map

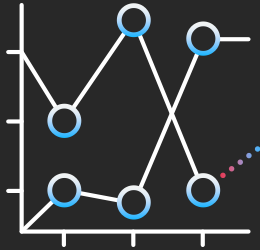


Amazon
CloudWatch logs

컨테이너는 현대화된
애플리케이션을 운영하기 위한
최고 도구입니다.

소프트웨어 딜리버리의 발전

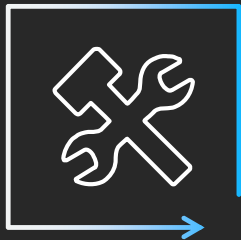
소프트웨어 제공을 위한 마이크로 서비스 FAQ



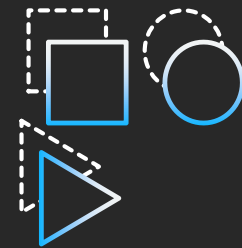
많은 서비스에 대한 릴리스
프로세스를 어떻게
관리합니까?



모범 사례를 어떻게
만들고 적용 합니까?

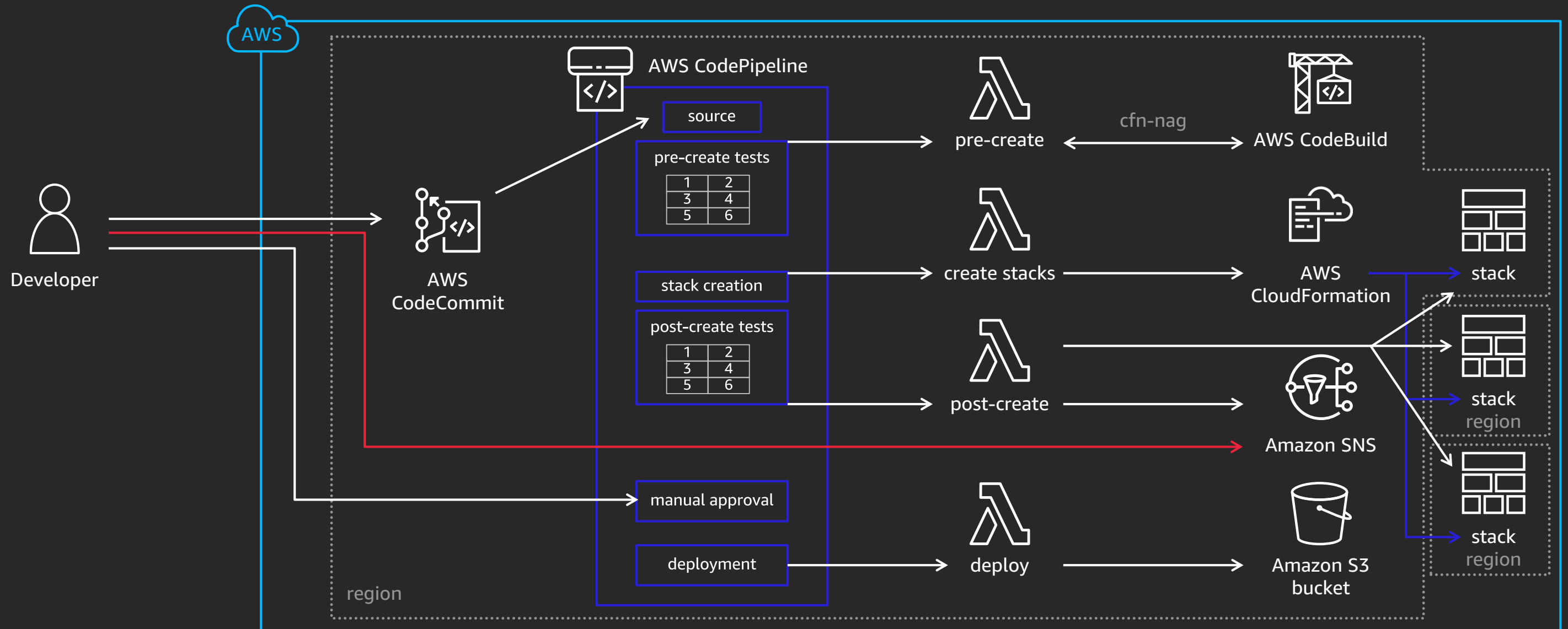


Lambda 애플리케이션을
작성하고 디버깅하려면
어떻게해야 합니까?

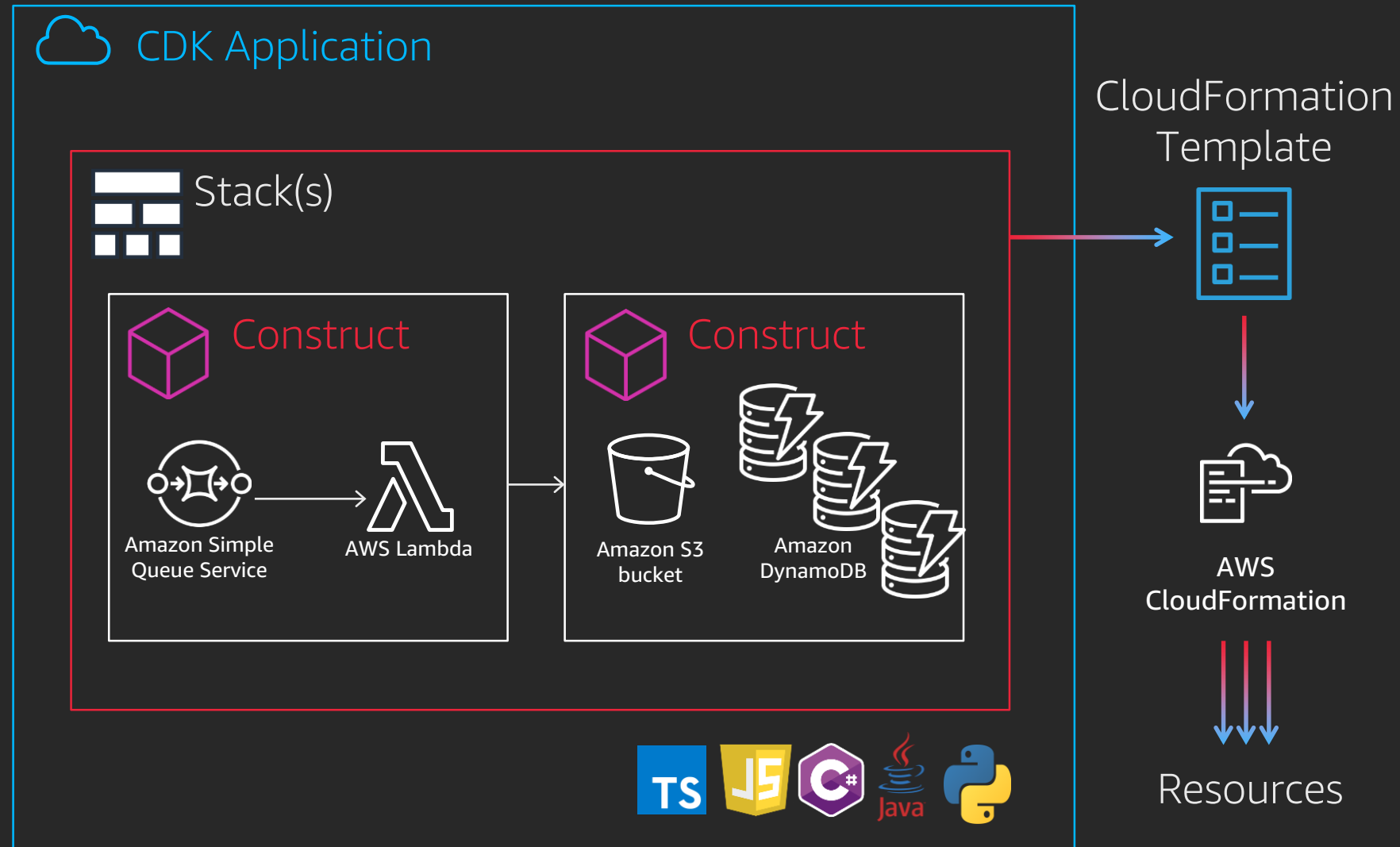


분산 아키텍처에서 임시
리소스를 모니터링하려면
어떻게해야 합니까?

인프라를 가장 잘 모델링하고 프로비저닝하려면 어떻게 해야 할까요?



AWS Cloud Development Kit

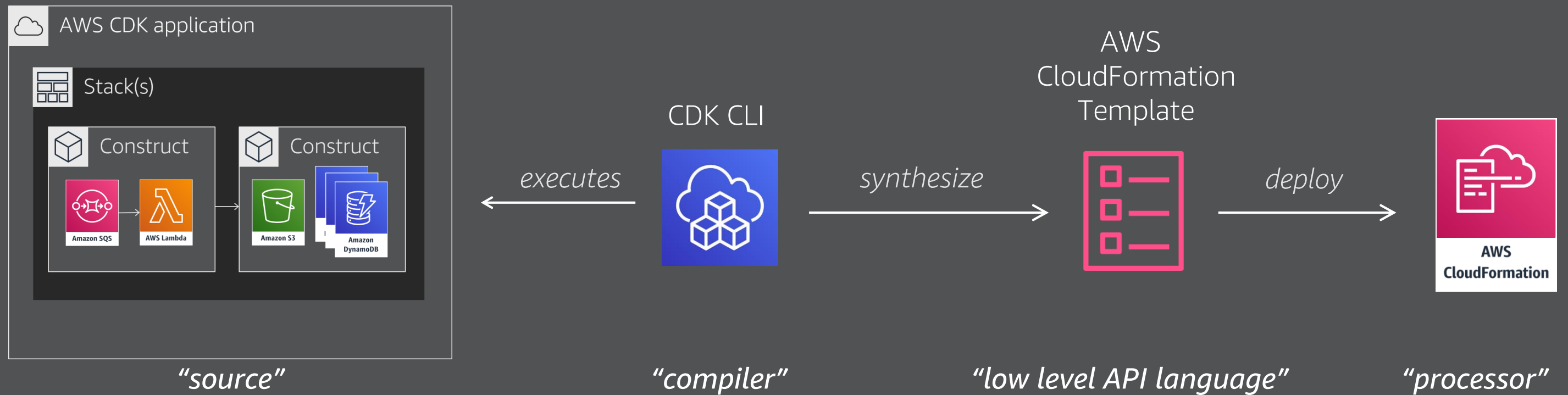


CDK: 애플리케이션 패키징

```
//  
// create static web site as S3 assets  
//  
var path = require('path');  
const asset = new assets.Asset(this, 'YourSampleApp', {  
    path: path.join(__dirname, '../html')  
});
```

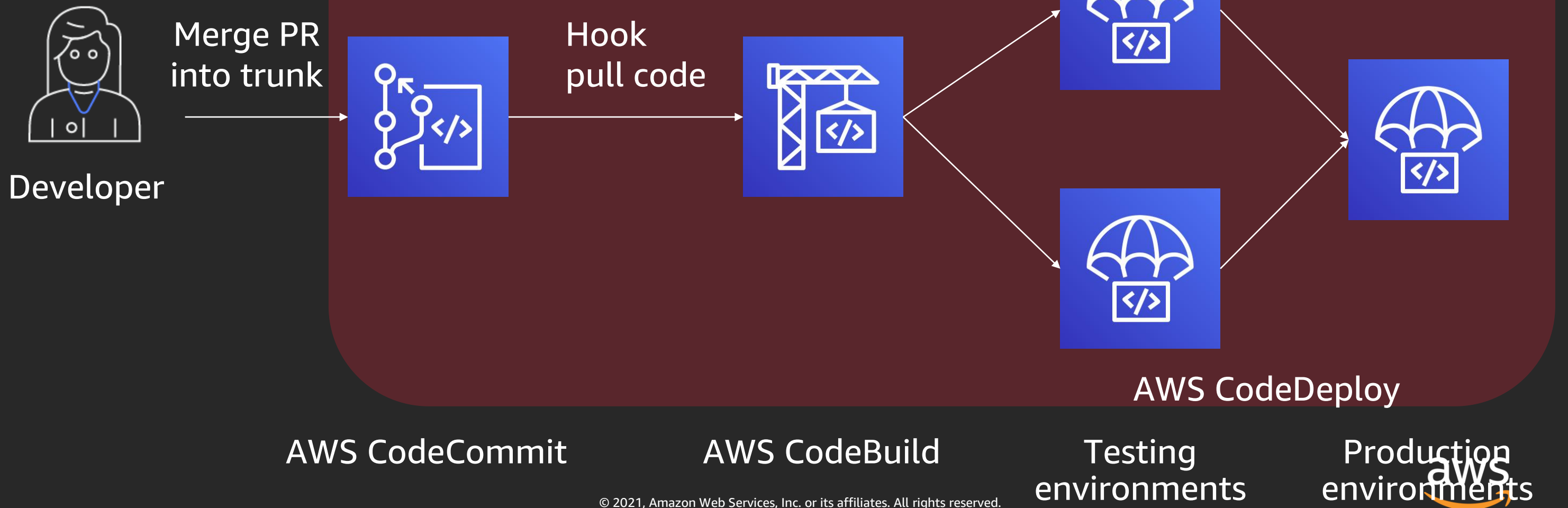
CDK: 개별 개발환경 배포

```
$ cdk deploy
```



지속적인 배포

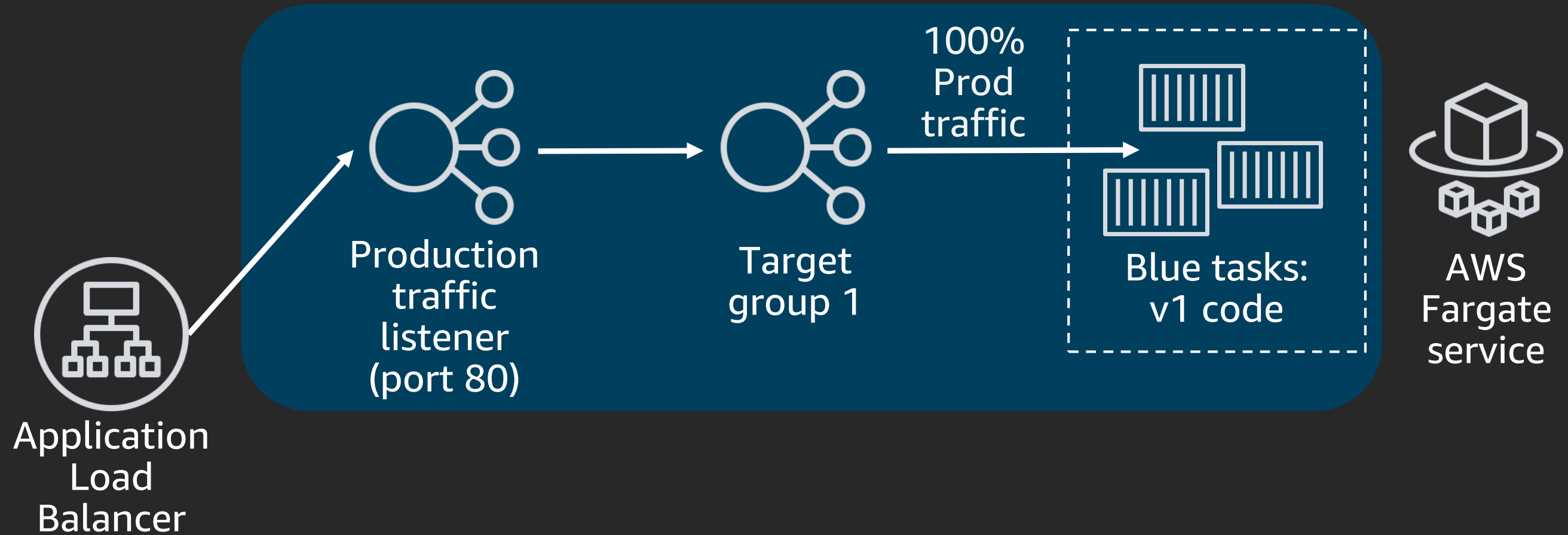
AWS CodePipeline



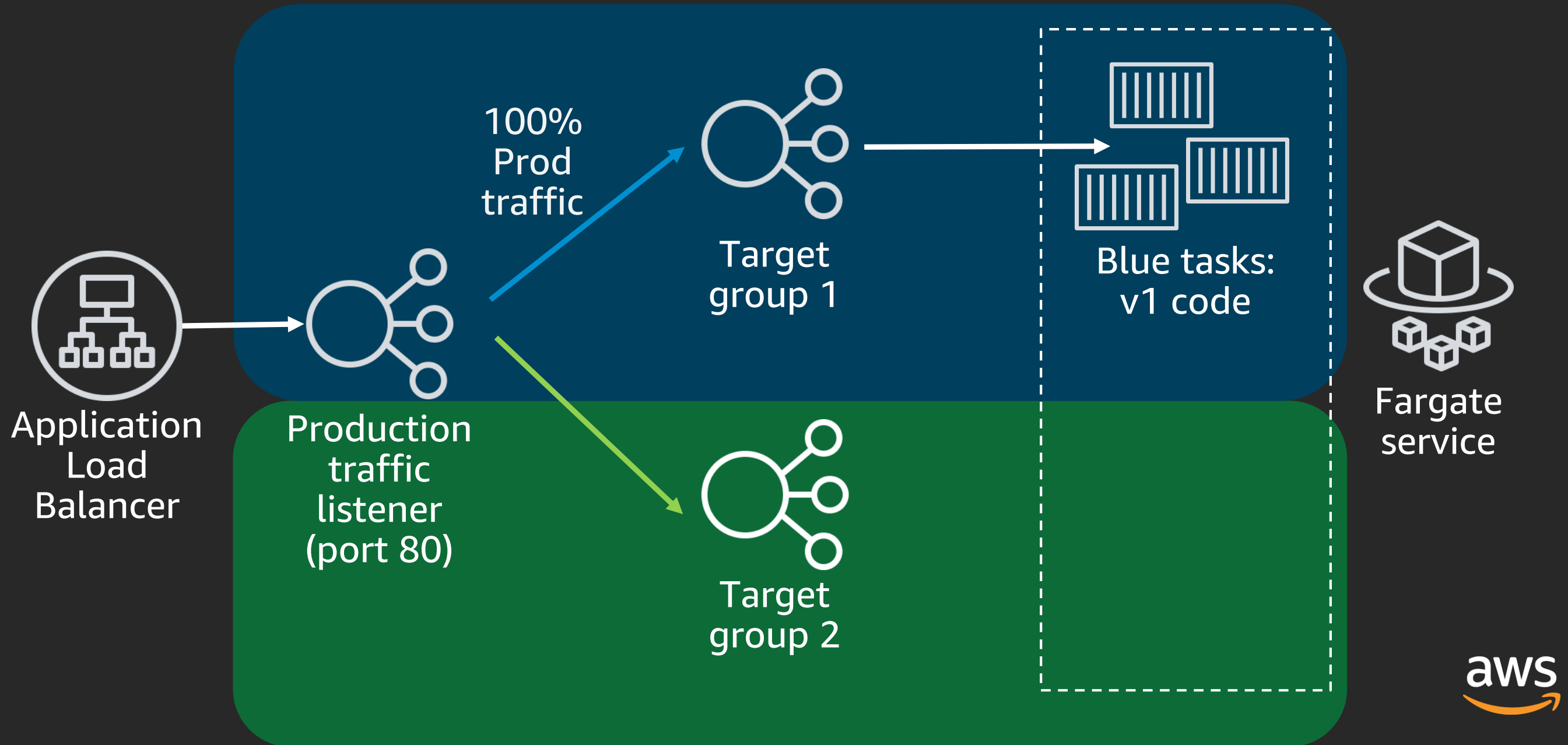
개발 인프라를 IaC 로 배포

```
// create the source action (github)
const sourceOutput = new pipeline.Artifact();
const sourceAction = new pipeline_actions.GitHubSourceAction({
  actionName: "GitHubTrigger",
  owner: github.owner,
  repo: github.repo,
  oauthToken: cdk.SecretValue.secretsManager(github.secret_manager_secret_name),
  output: sourceOutput,
  branch: 'master'
});
```

블루-그린 배포

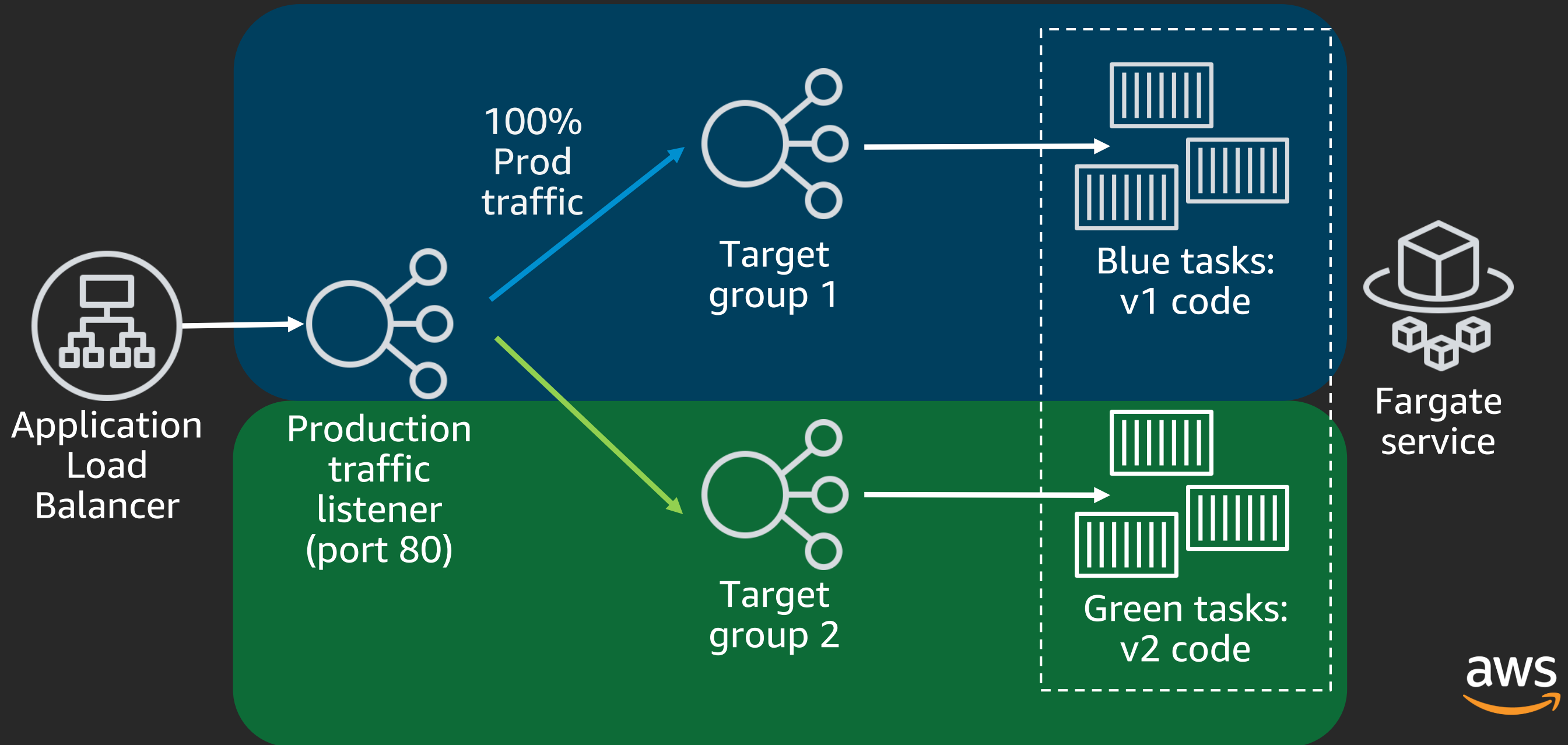


블루-그린 배포



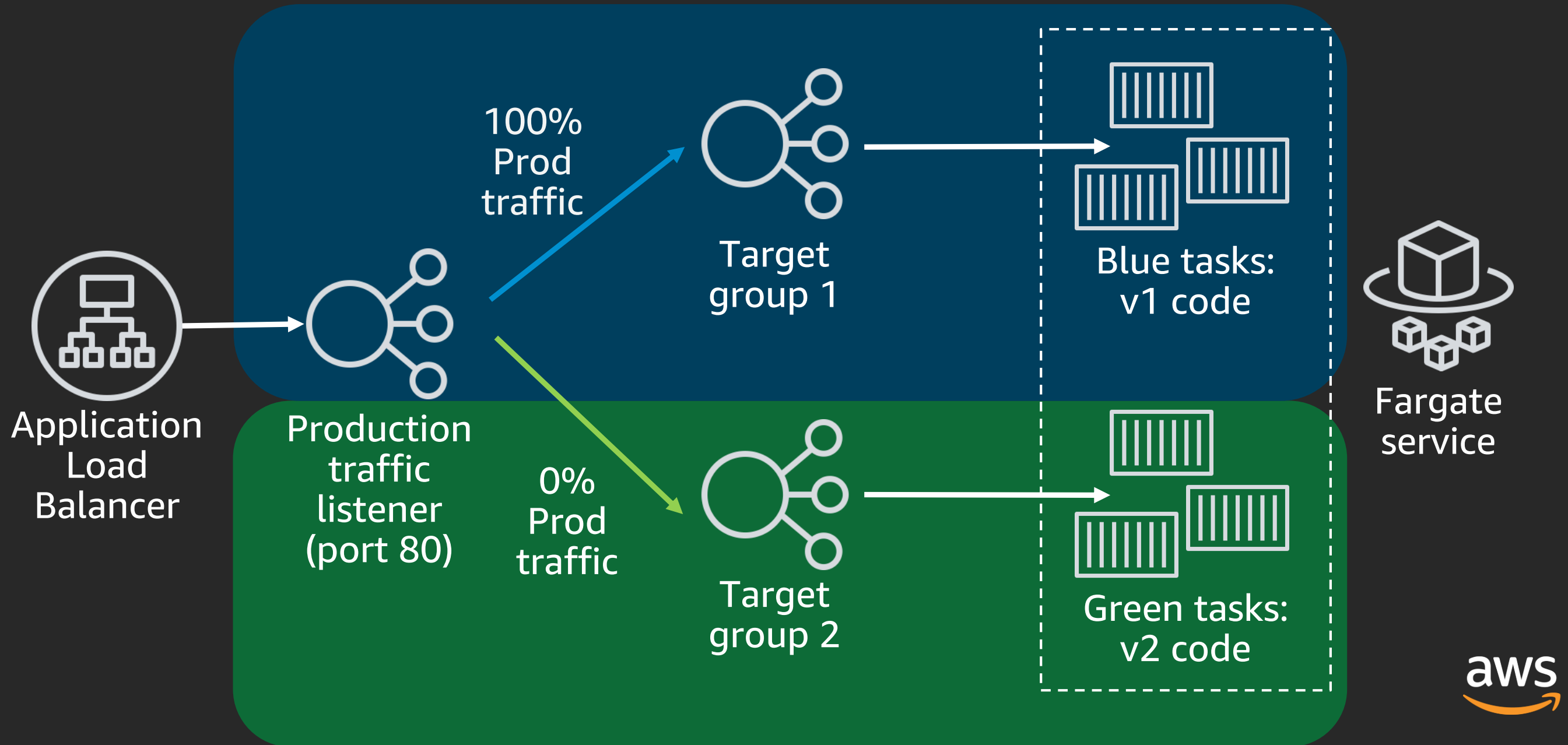
블루-그린 배포

그린 테스트 프로비전



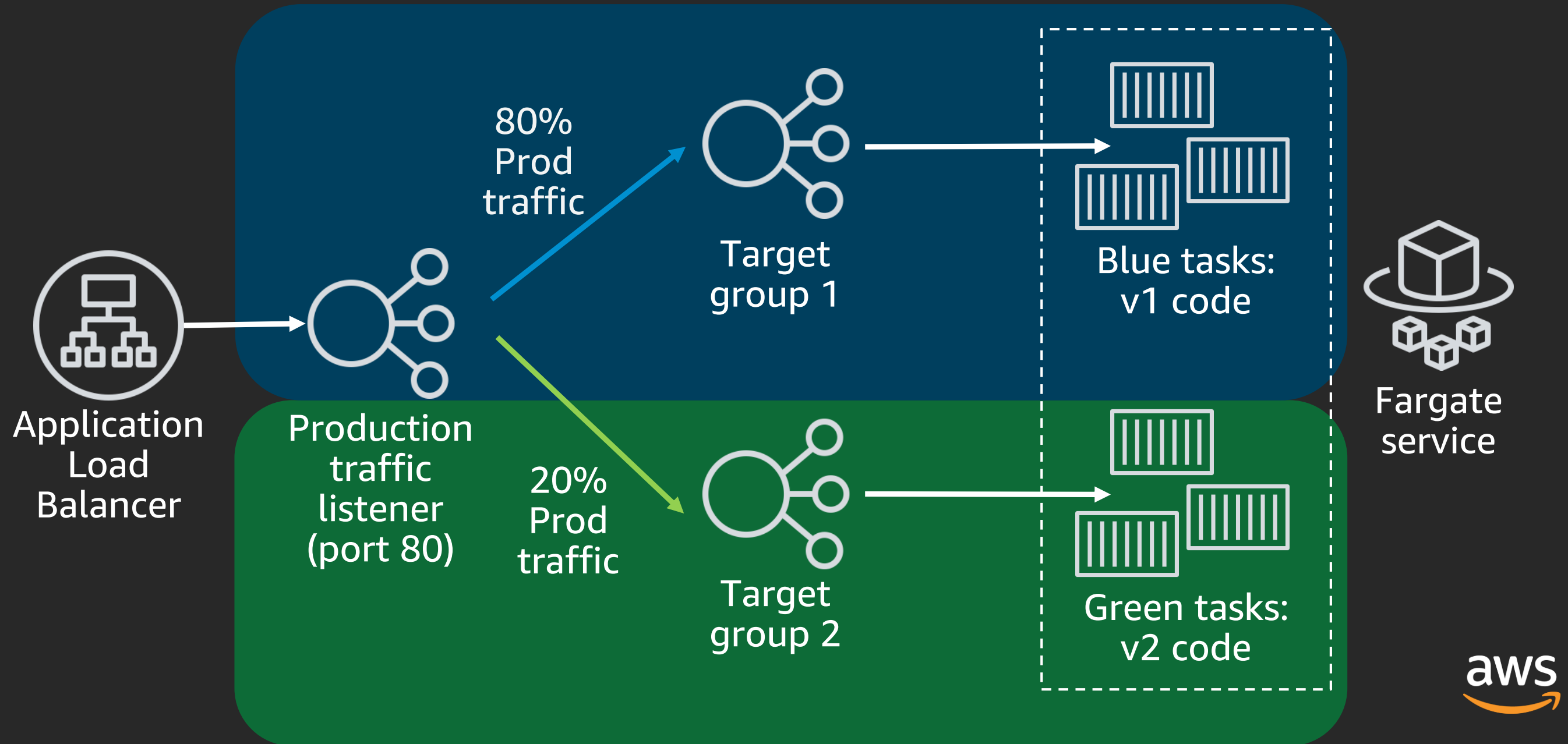
블루-그린 배포

그린 테스트가 prod 트래픽을 수신하기 전에 테스트 엔드 포인트에 대해 후크



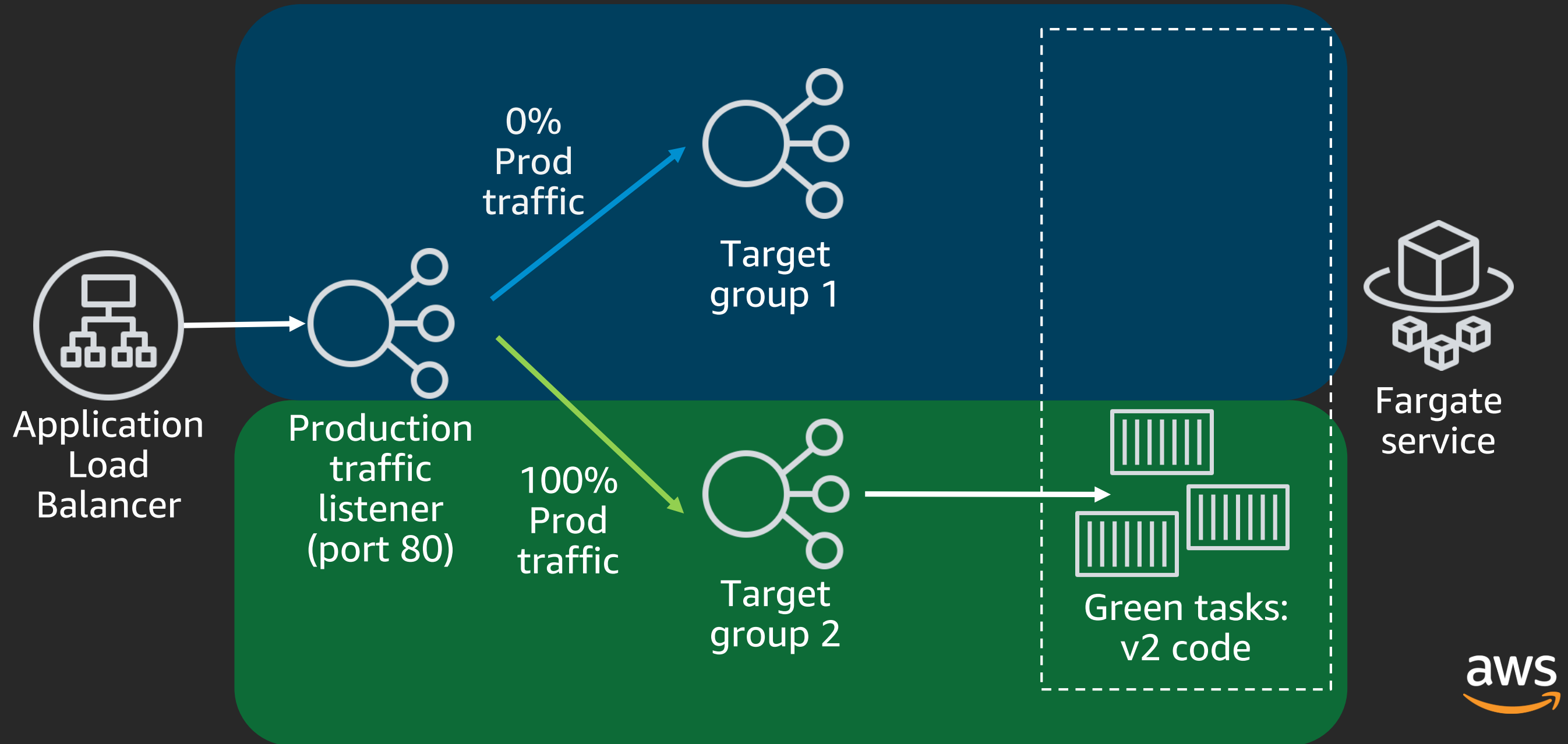
블루-그린 배포

그린 테스트로 트래픽 전환, 장애 발생시 롤백



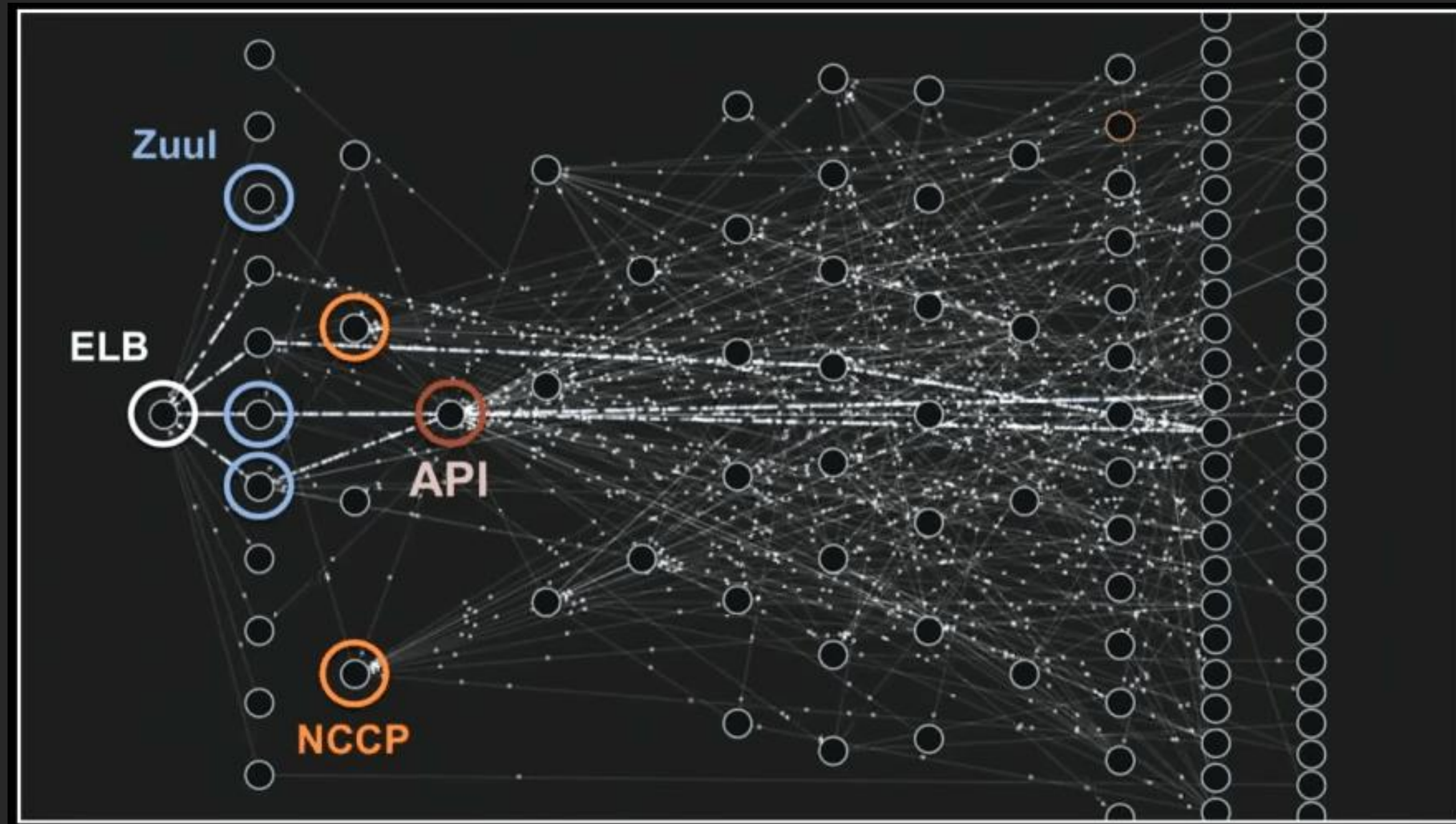
블루-그린 배포

블루 테스트 제거



고급 DevOps 도구

Netflix Guide to Microservices - infoQ



대규모 DevOps 운영 과제

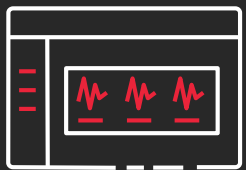
Operational challenges at scale



DevOps 파이프라인에서 발생하는 데이터 볼륨이 크고 단계별로 서로 이질적인 패턴



문제가 발생하면 사용자는 데이터 소스와 도구를 수동으로 상호 연결하는 데 많은 시간과 노력을 소비



새로운 서비스가 채택됨에 따라 새로운 경보 및 모니터링 절차를 구성하려면 심층적인 DevOps / CloudOps 전문 지식이 필요



여러 도구의 경보 및 알림으로 인해 경보 피로가 발생하고 가장 중요한 문제를 식별하지 못함

Amazon DevOps Guru

애플리케이션 가용성을 향상시키는 ML 기반 클라우드 운영 서비스



DevOps Guru는 ML 기반 서비스로, 개발자와 운영자가 문제를 자동으로 감지하여 애플리케이션 가용성을 개선하고 비용이 많이 드는 다운타임을 줄일 수 있습니다. 머신 러닝 경험이 필요하지 않습니다.

Why Amazon DevOps Guru?



손쉬운 사용법

No configuration or ML experience required



운영 문제 자동 감지

DevOps Guru continuously analyzes streams of disparate data and metrics to determine application behavior



빠른 해결

Helps to resolve issues quickly with ML-powered insights and recommendations



확장성

Easily scales and maintains availability as new AWS workloads are added



노이즈 억제

Helps overcome alarm fatigue by automatically correlating and grouping related anomalies

Amazon DevOps Guru

Leverage Amazon DevOps Guru's machine learning capabilities

- 심각도가 높은 사고 이상을 감지하고 노이즈를 제거합니다.
- 장애에 의해 트리거 된 관련 메트릭의 이상, 추적, 변경 및 로그 이벤트를 상호 연관시킵니다.
- 상황 별 데이터로 장애를 인식
- 근본 원인이 될만한 가능성을 식별
- 문제를 더 빨리 해결하도록 지원

DevOps Guru 적용 방법

Amazon DevOps Guru (Public Preview)

전체 계정 또는 특정 CFN 스택
선택



Account

OR



CloudFormation
Stack

범위 선택

DevOps Guru는 통계 / 로그
분석을 시작하여 통찰력을 생성

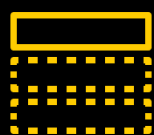
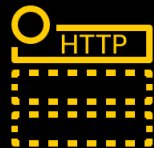


인사이트 생성

OpsCenter, SNS 및 타사
플랫폼과 통합



SNS



워크플로우와 통합



결론

The AWS Developer Tools

*Accenture 2018 개발자 에코 시스템 설문 조사에서 AWS를
가장 개발자 친화적이고 가장 미래 지향적 인 것으로 선정*

**다양한 엔터프라이즈
급 도구 세트**

AWS에는 광범위한 사용 사례를
대규모로 지원하도록 설계된 강력한
도구 세트

**현대화 애플리케이션
개발을 위한 구축
솔루션**

최신 애플리케이션 개발 기술 (예 : CI /
CD, IaC) 및 기술 (예 : 서버리스,
컨테이너)을 지원하도록 설계

Helpful Links

Developer Tools page: <https://aws.amazon.com/products/developer-tools>

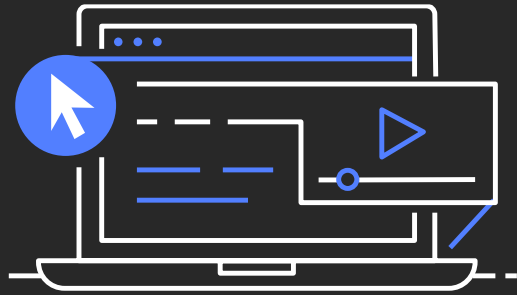
AWS Developer Center: <https://aws.amazon.com/developer>

Developer Tools/DevOps Blog: <https://aws.amazon.com/blogs/devops>

More on DevOps: <https://aws.amazon.com/devops>

DevOps Guru: <https://aws.amazon.com/devops-guru/>

AWS 디지털 교육



Flexibility to Learn Your Way

550개 이상의
무료 디지털 교육 및
심층적 강의실 교육을 통해
클라우드 기술 역량을
업그레이드 하세요!

주요 교육

- [AWS Cloud Practitioner Essentials \(Second Edition\)](#)
AWS 클라우드 기초에 대해 학습하고, 기초 자격증인 AWS Certified Cloud Practitioner 시험을 준비할 수 있습니다.
- [AWS Security Fundamentals](#)
AWS 액세스 제어 및 관리, 거버넌스, 로깅, 그리고 암호화 방법을 포함한 기본적인 클라우드 컴퓨팅 및 AWS 보안 개념에 대해 알아봅니다.
- [AWS Database Offerings](#)
데이터베이스 기술 및 아키텍처에 대한 기본 개요를 제공하며, 다양한 AWS 데이터베이스 서비스를 소개합니다.
- [Data Analytics Fundamentals](#)
다양한 데이터 분석 프로세스에 대해 학습하며, 데이터 분석 솔루션을 구축하고 개선하는 데 도움이 되는 AWS 서비스 및 솔루션을 소개합니다.

AWS Builders Online Series에 참석해주셔서 대단히 감사합니다.

저희가 준비한 내용, 어떻게 보셨나요?
더 나은 세미나를 위하여 **설문을 꼭 작성해 주시기 바랍니다.**



aws-korea-marketing@amazon.com



twitter.com/AWSKorea



facebook.com/amazonwebservices.ko



youtube.com/user/AWSKorea



slideshare.net/awskorea



twitch.tv/aws



Builders Online Series

Thank you