



SUMMIT  
ONLINE

# CI/CD at scale: Best practices with AWS DevOps services

Loh Yiang Meng  
Solutions Architect  
Amazon Web Services

Martin Lim  
Chief Executive Officer  
Electrify Asia

Arshad Zackeriya  
Senior DevOps Engineer  
Electrify Asia

# Agenda

- What is DevOps?
- Pipeline automation
- Safe deployments
- Repeatable infrastructure changes
- CI/CD @ Electrify Asia
- Demo

# What is DevOps?

DevOps =

# What is DevOps?

DevOps = Culture + Practices + Tools

# What is DevOps?

DevOps = Culture + Practices + Tools

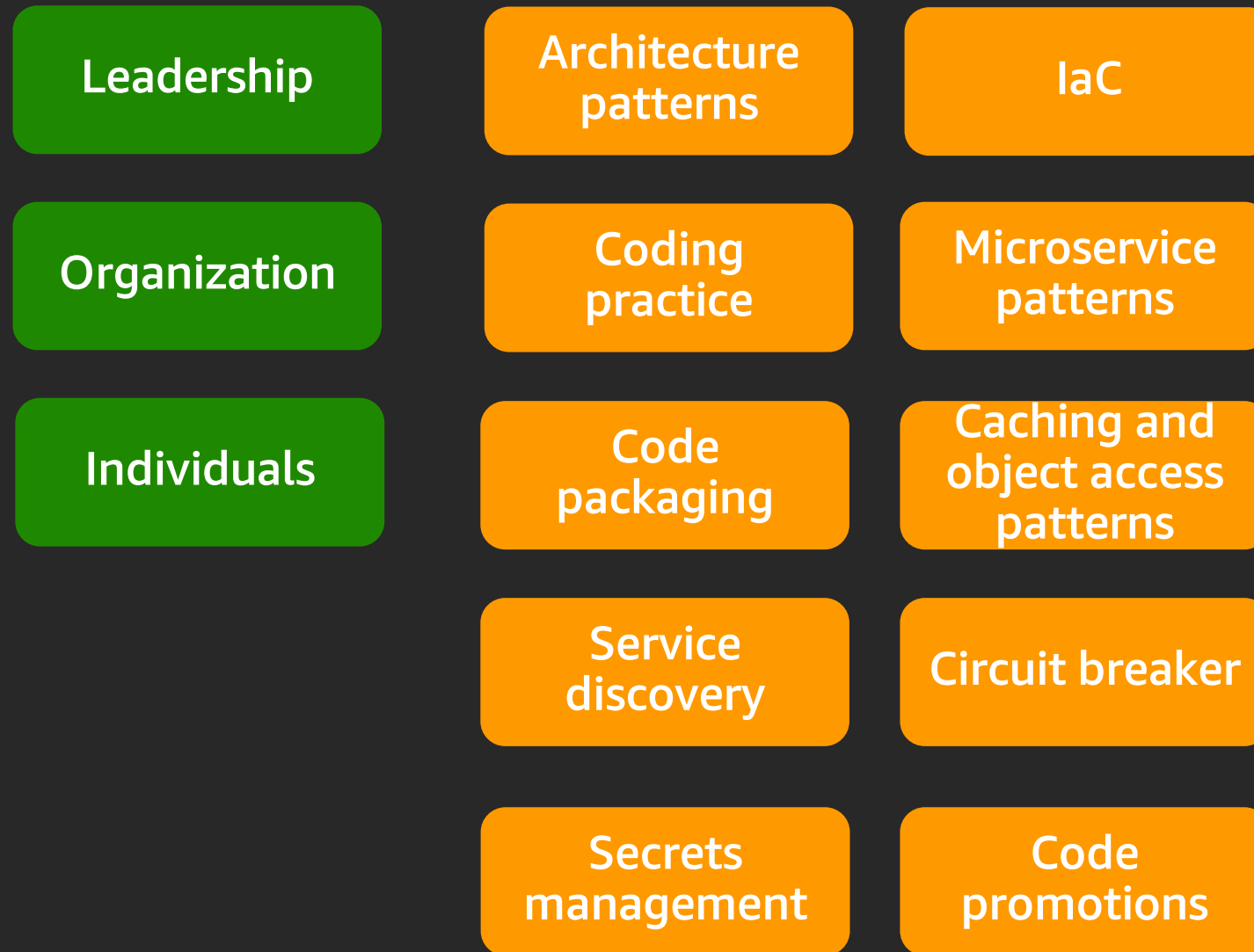
Leadership

Organization

Individuals

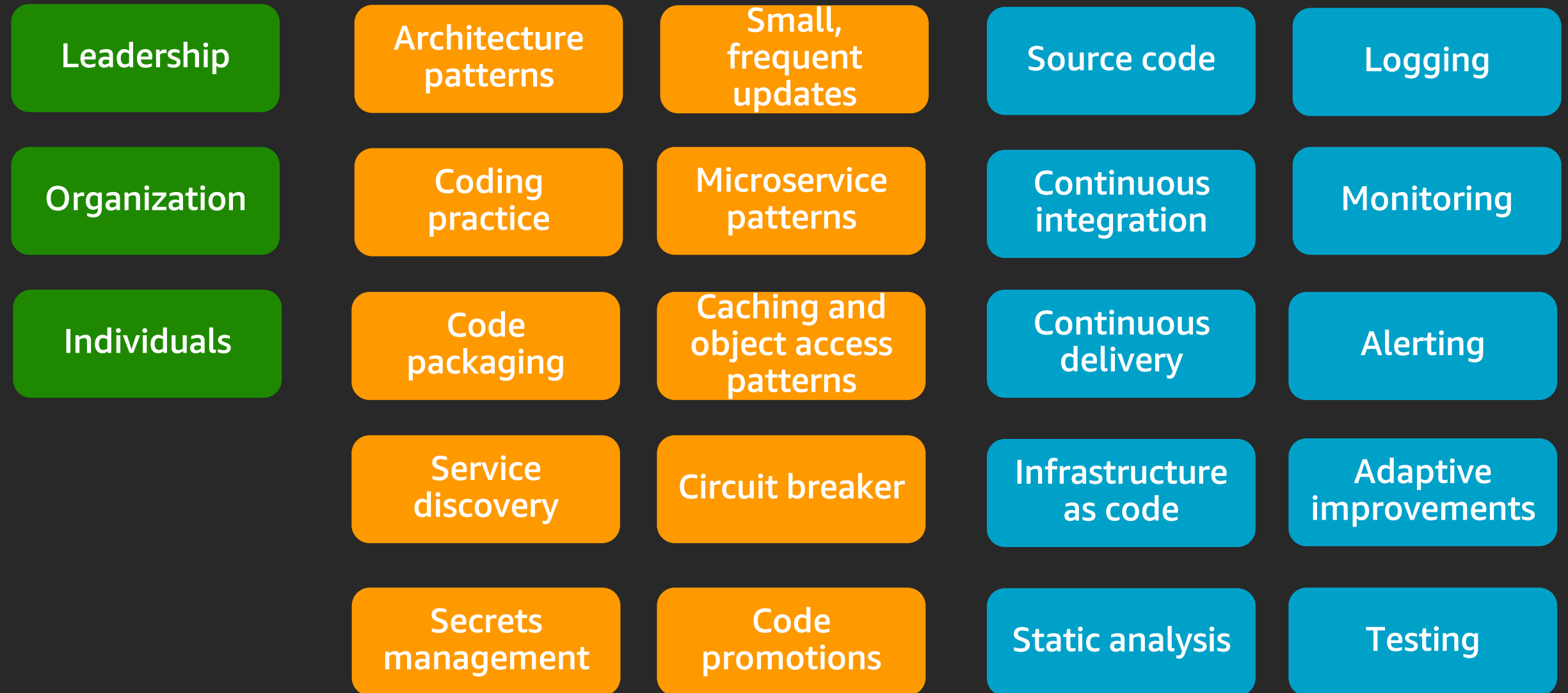
# What is DevOps?

DevOps = Culture + Practices + Tools



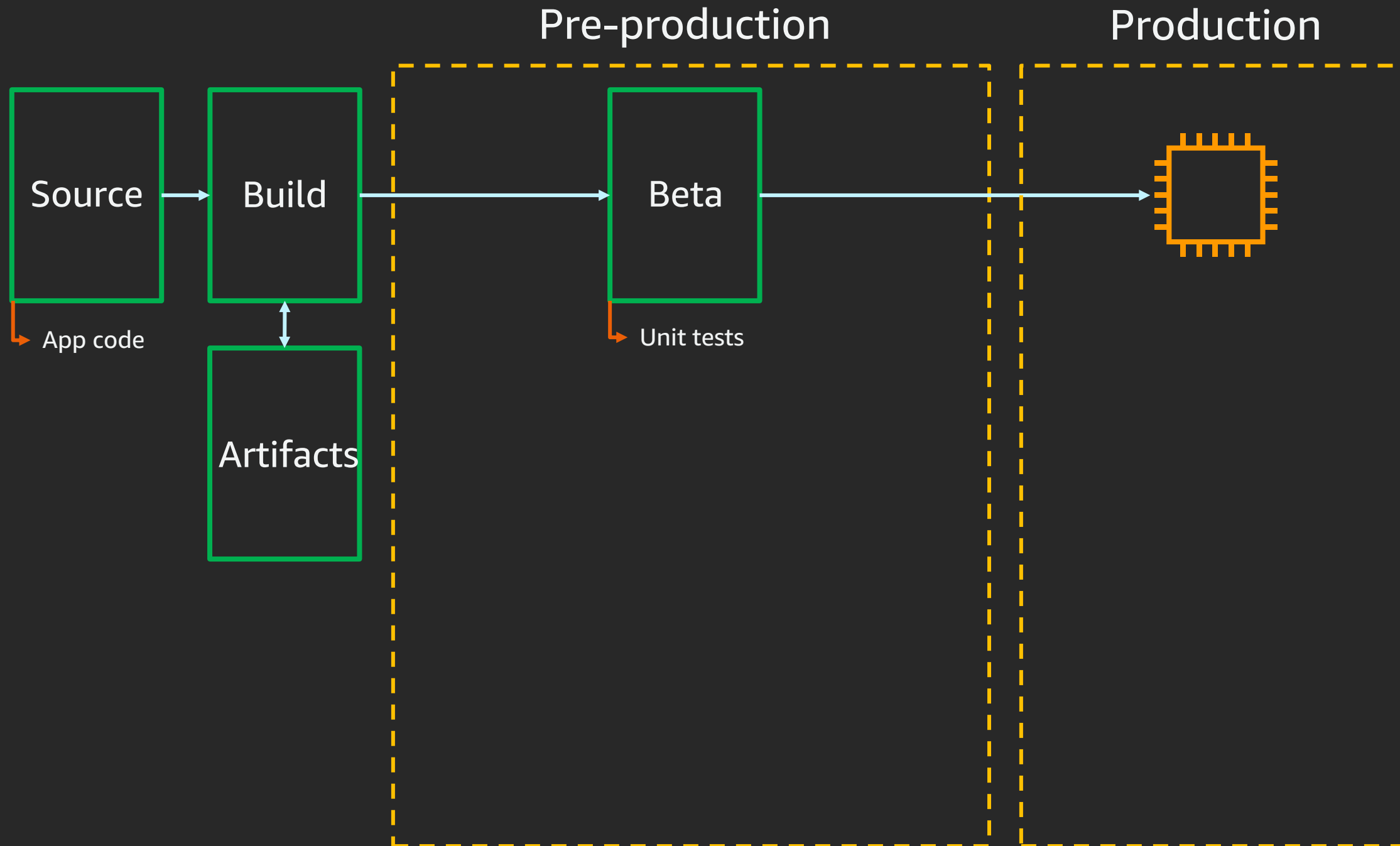
# What is DevOps?

DevOps = Culture + Practices + Tools

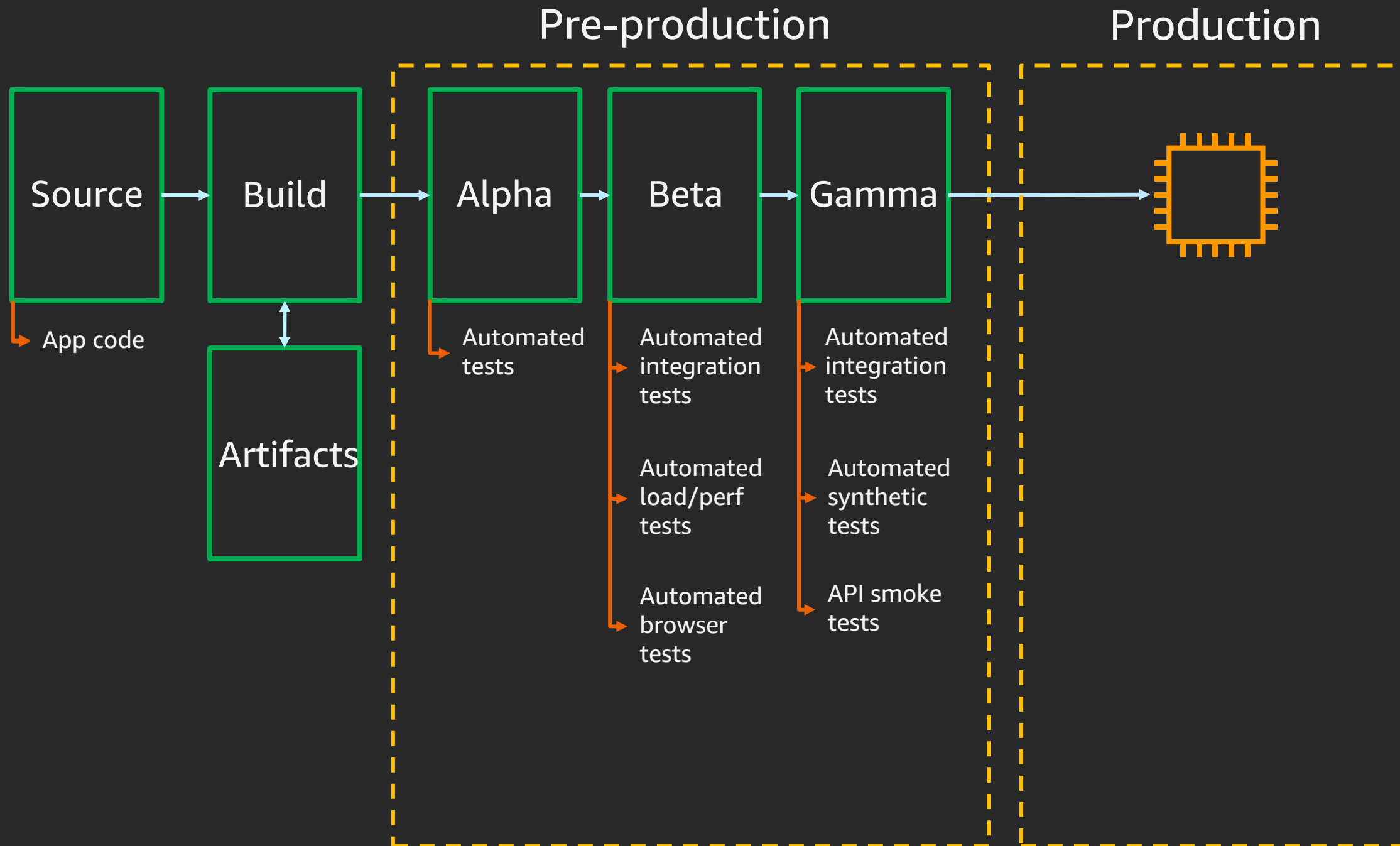




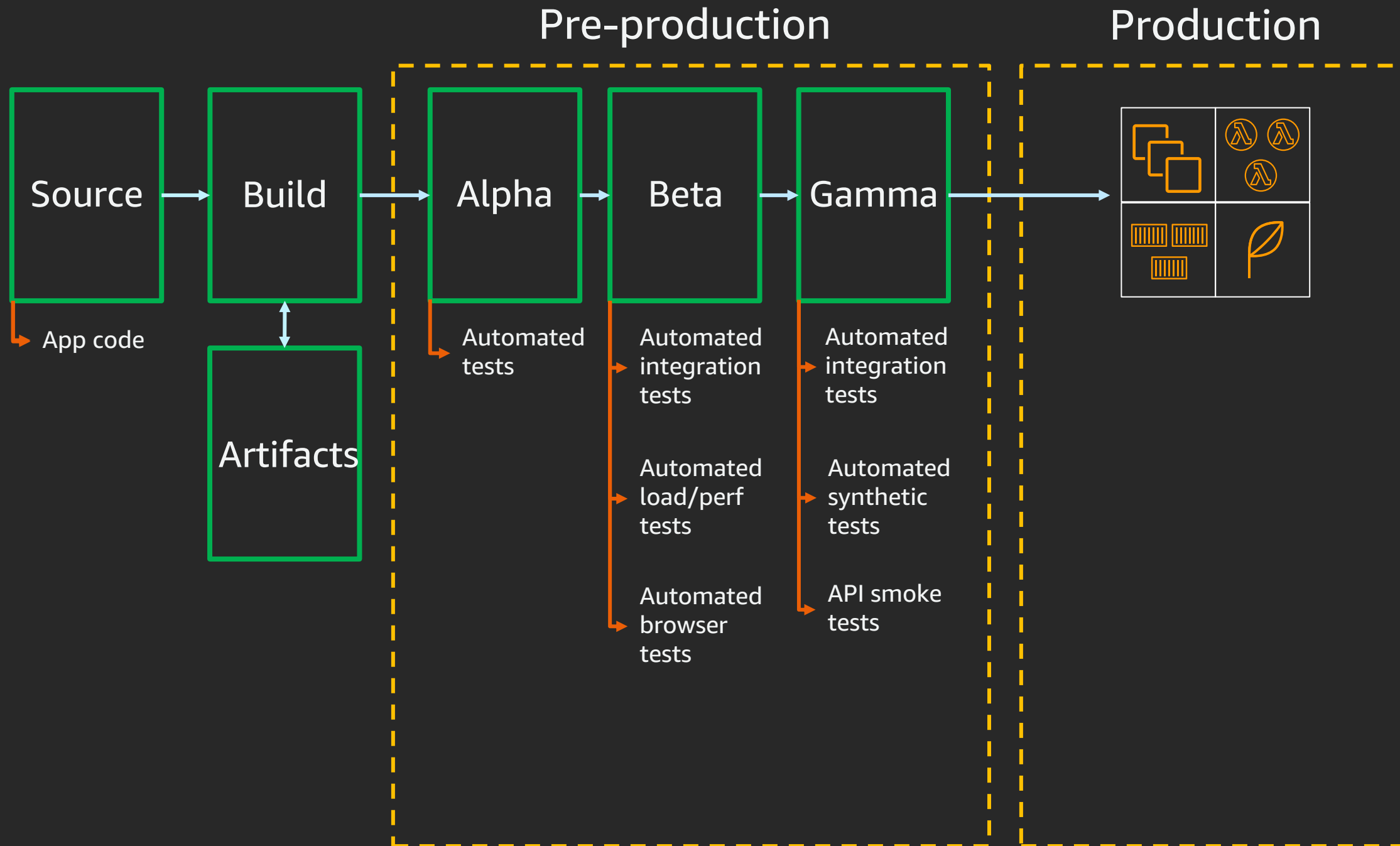
# What is DevOps at scale?



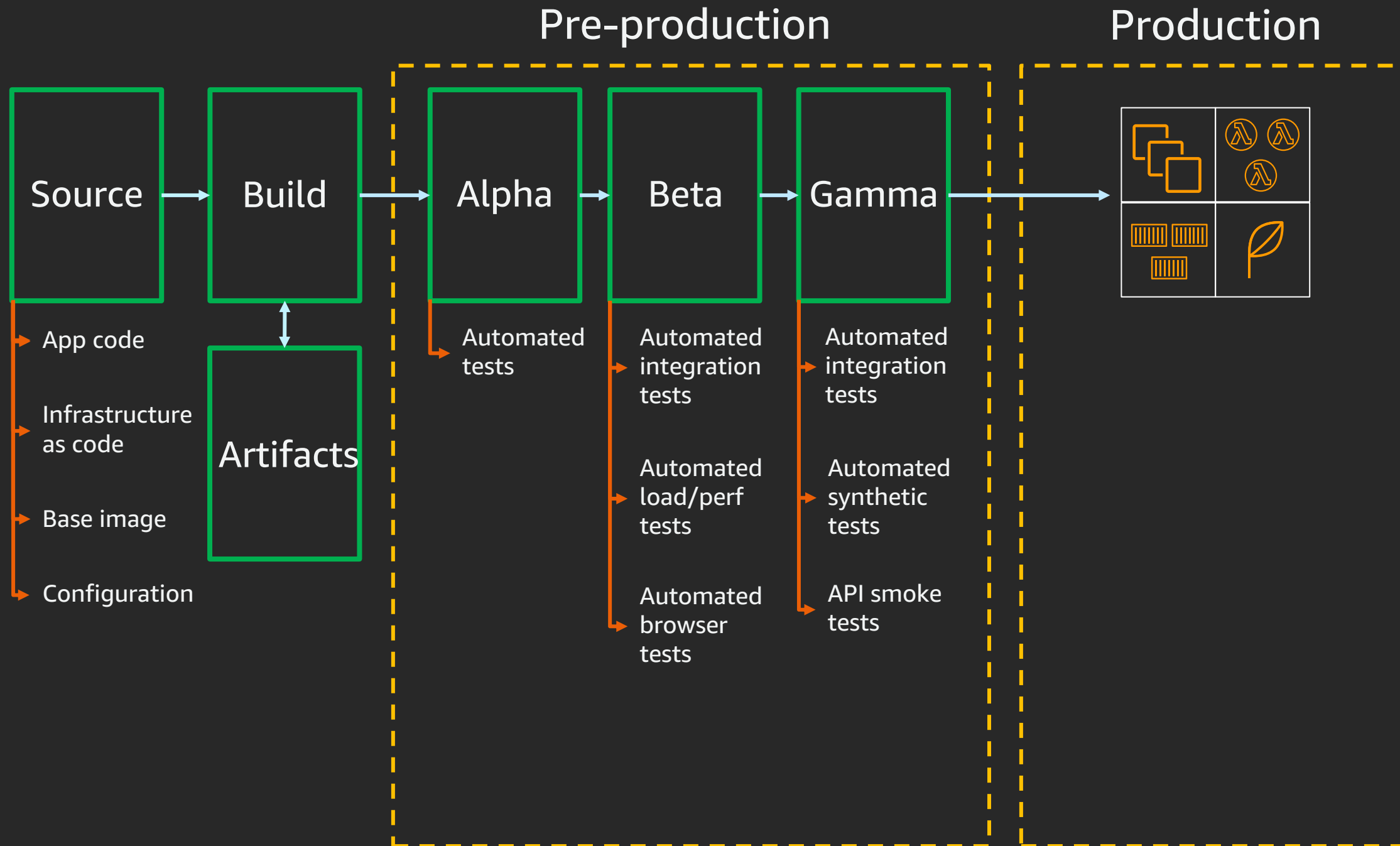
# What is DevOps at scale?



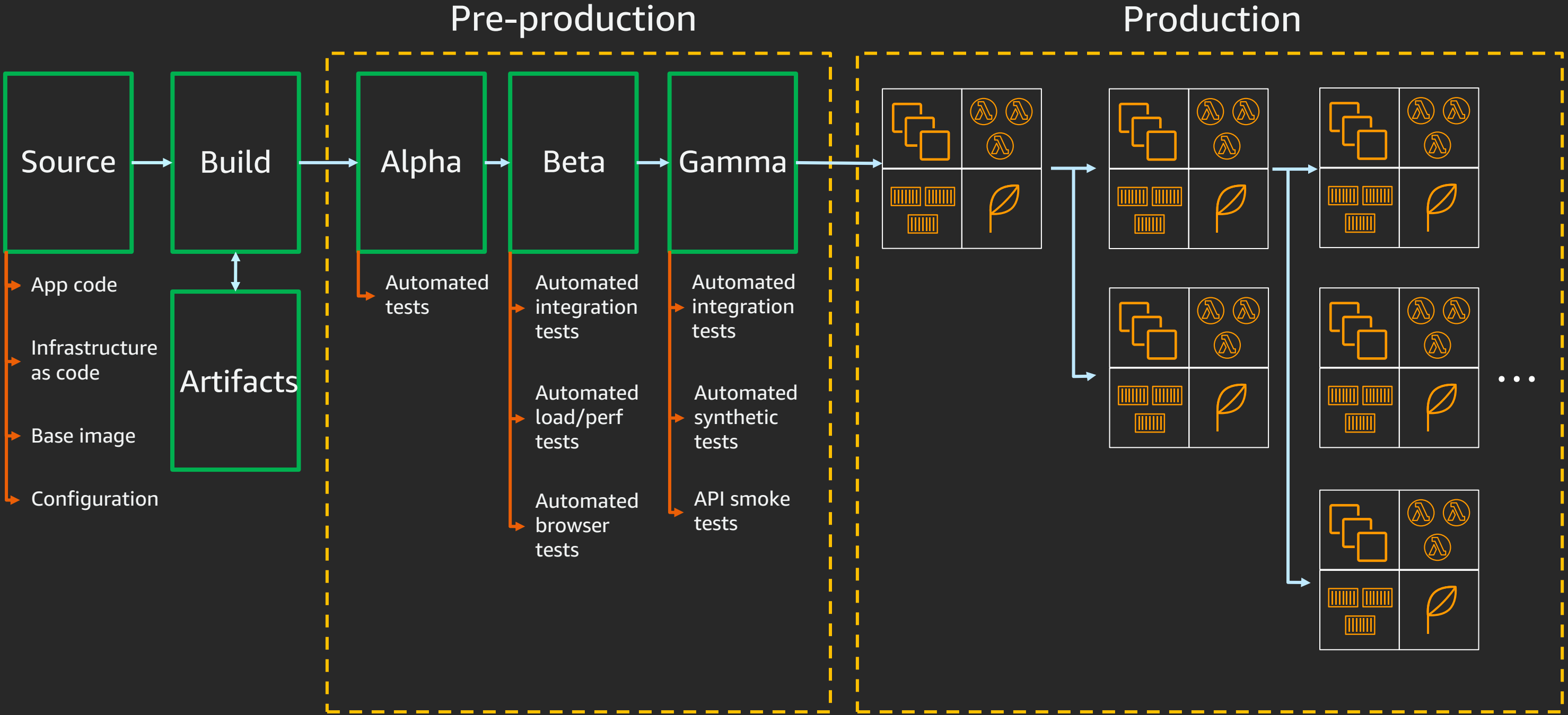
# What is DevOps at scale?



# What is DevOps at scale?



# What is DevOps at scale?



# Best practices for CI/CD

1

Pipeline  
automation

2

Safe  
deployments

3

Repeatable  
infrastructure  
changes

# Best practices for CI/CD

1

Pipeline  
automation

2

Safe  
deployments

3

Repeatable  
infrastructure  
changes

# Release process stages



- Check in source code such, as .java files
- Peer review new code

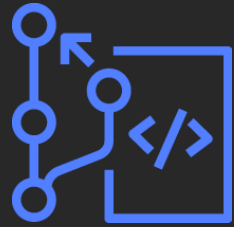
- Compile code
- Unit tests
- Style checkers
- Create container images and deployment packages

- Integration tests with other systems
- Load testing
- UI tests
- Security testing

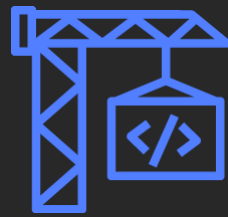
- Deploy to production environments
- Monitor code in production in order to quickly detect errors



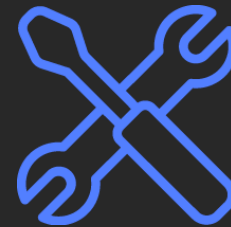
# AWS code services



AWS CodeCommit



AWS CodeBuild



AWS CodeBuild  
+ third-party  
tooling



AWS CodeDeploy



AWS CodePipeline

# AWS CodePipeline



- Managed continuous delivery service
- Model and visualize release process
- Automated pipeline trigger on code change
- Integrates with third-party tools

# AWS CodePipeline: Supported sources

## Via branch

AWS CodeCommit

GitHub

★ Bitbucket

## Via object/folder

Amazon Simple  
Storage Service  
(Amazon S3)

## Via Docker image

Amazon Elastic  
Container Registry  
(Amazon ECR)

# AWS CodePipeline: Supported triggers

Automatically kick off release

## Amazon EventBridge

- Scheduled (nightly release)
- AWS Health events (AWS Fargate platform retirement)

Available in Amazon EventBridge console, API, SDK, CLI, and AWS CloudFormation

## Webhooks

- Docker Hub
- Quay
- Artifactory

Available in AWS CodePipeline API, SDK, CLI, and AWS CloudFormation

# AWS CodePipeline: Supported deployment targets

## Amazon EC2

AWS CodeDeploy

AWS Elastic Beanstalk

AWS OpsWorks Stacks

## Containers

AWS CodeDeploy

Amazon ECS

AWS Fargate

## Serverless

AWS CodeDeploy

AWS CloudFormation  
(AWS SAM)

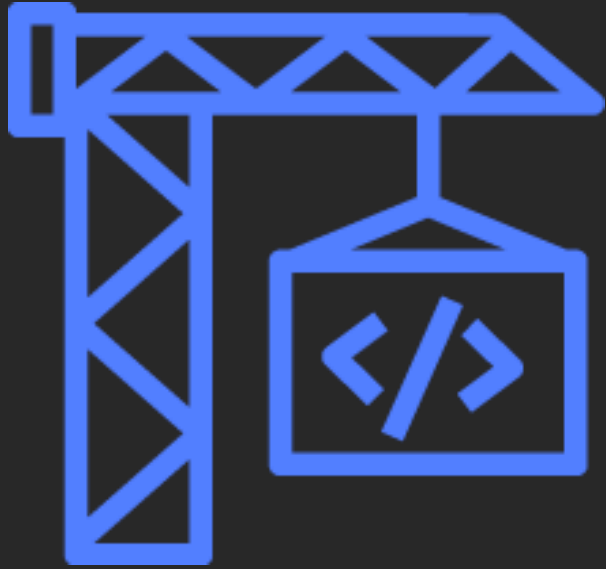
AWS Lambda

# Continuous integration goals



1. Automatically kick off a new build when new code is checked in
2. Build and test code in a consistent, repeatable environment
3. Continually have an artifact ready for deployment
4. Continually close feedback loop when build fails

# AWS CodeBuild



- Fully managed build service
- Isolated build containers for consistent, immutable environment
- Docker and AWS CLI out of box
- Ability to customize build environment

# AWS CodeBuild

```
version: 0.2
```

```
env:
```

```
  variables:
```

```
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
```

```
phases:
```

```
  install:
```

```
    runtime-versions:
```

```
      java: corretto8
```

```
  build:
```

```
    commands:
```

```
      - echo Build started on `date`  
      - mvn install
```

```
  post_build:
```

```
    commands:
```

```
      - echo Test started on `date`  
      - mvn surefire-report:report
```

```
reports:
```

```
  SurefireReports:
```

```
    files:
```

```
      - '**/*'
```

```
    base-directory: 'target/surefire-reports'
```

```
artifacts:
```

```
  type: zip
```

```
  files:
```

```
    - target/messageUtil-1.0.jar
```

```
  discard-paths: yes
```

} Variables to be used by phases of build

} Execute build command

} Execute unit tests

} Create and store build artifacts in Amazon S3



# AWS CodeBuild

```
version: 0.2
```

```
env:
```

```
  variables:
```

```
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
```

```
phases:
```

```
  install:
```

```
    runtime-versions:
```

```
      java: corretto8
```

```
  build:
```

```
    commands:
```

```
      - echo Build started on `date`  
      - mvn install
```

```
  post_build:
```

```
    commands:
```

```
      - echo Test started on `date`  
      - mvn surefire-report:report
```

```
reports:
```

```
  SurefireReports:
```

```
    files:
```

```
      - '**/*'
```

```
    base-directory: 'target/surefire-reports'
```

```
artifacts:
```

```
  type: zip
```

```
  files:
```

```
    - target/messageUtil-1.0.jar
```

```
  discard-paths: yes
```

★  
} v0.1 – each build cmd in separate shell  
} v0.2 – each build cmd in same shell

} Variables to be used by phases of build

} Execute build command

} Execute unit tests

} Create and store build artifacts in Amazon S3

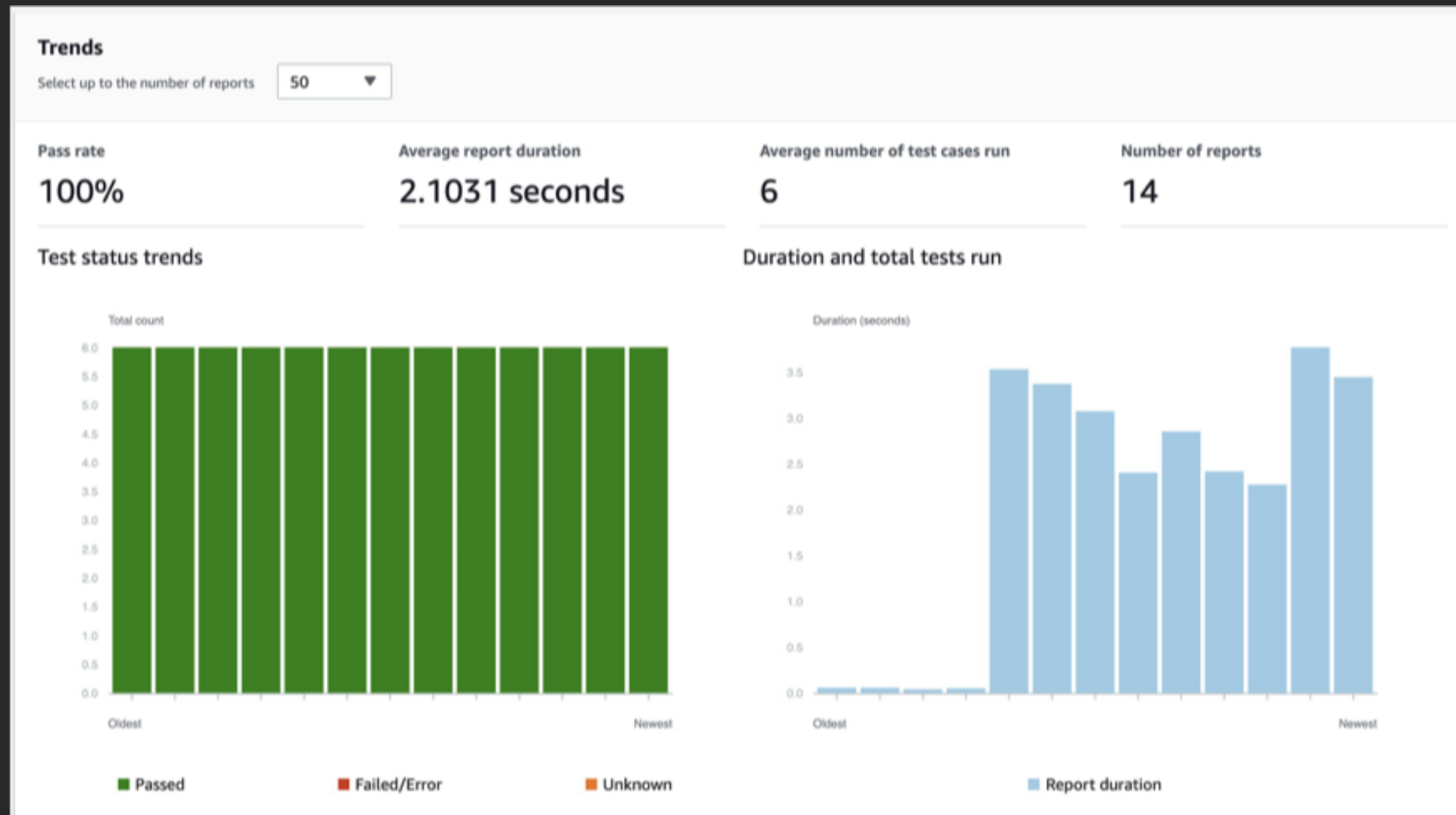
# AWS CodeBuild

```
version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
phases:
  install:
    runtime-versions:
      java: corretto8
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
    commands:
      - echo Test started on `date`
      - mvn surefire-report:report
reports:
  SurefireReports:
    files:
      - '**/*'
    base-directory: 'target/surefire-reports'
artifacts:
  type: zip
  files:
    - target/messageUtil-1.0.jar
discard-paths: yes
```

- ★ v0.1 – each build cmd in separate shell
- v0.2 – each build cmd in same shell
- Variables to be used by phases of build
- Execute build command
- Execute unit tests
- ★ Reports output location
- Create and store build artifacts in Amazon S3

# AWS CodeBuild



See breakdown of individual unit tests, status of the tests, duration, and messages from the tests

# Best practices for CI/CD

1

Pipeline  
automation

2

Safe  
deployments

3

Repeatable  
infrastructure  
changes

# Continuous deployment goals



1. Automatically deploy new changes to staging environments for testing
2. Deploy to production safely without impacting customers
3. Deliver to customers faster: Increase deployment frequency and reduce change lead time and change failure rate

# AWS CodeDeploy



- Automates code deployments
- Handles complexity of application updates
- Avoid downtime during deployment
- Roll back automatically upon failure
- Limit “blast radius” with traffic control

# AWS CodeDeploy: Amazon EC2 deployments

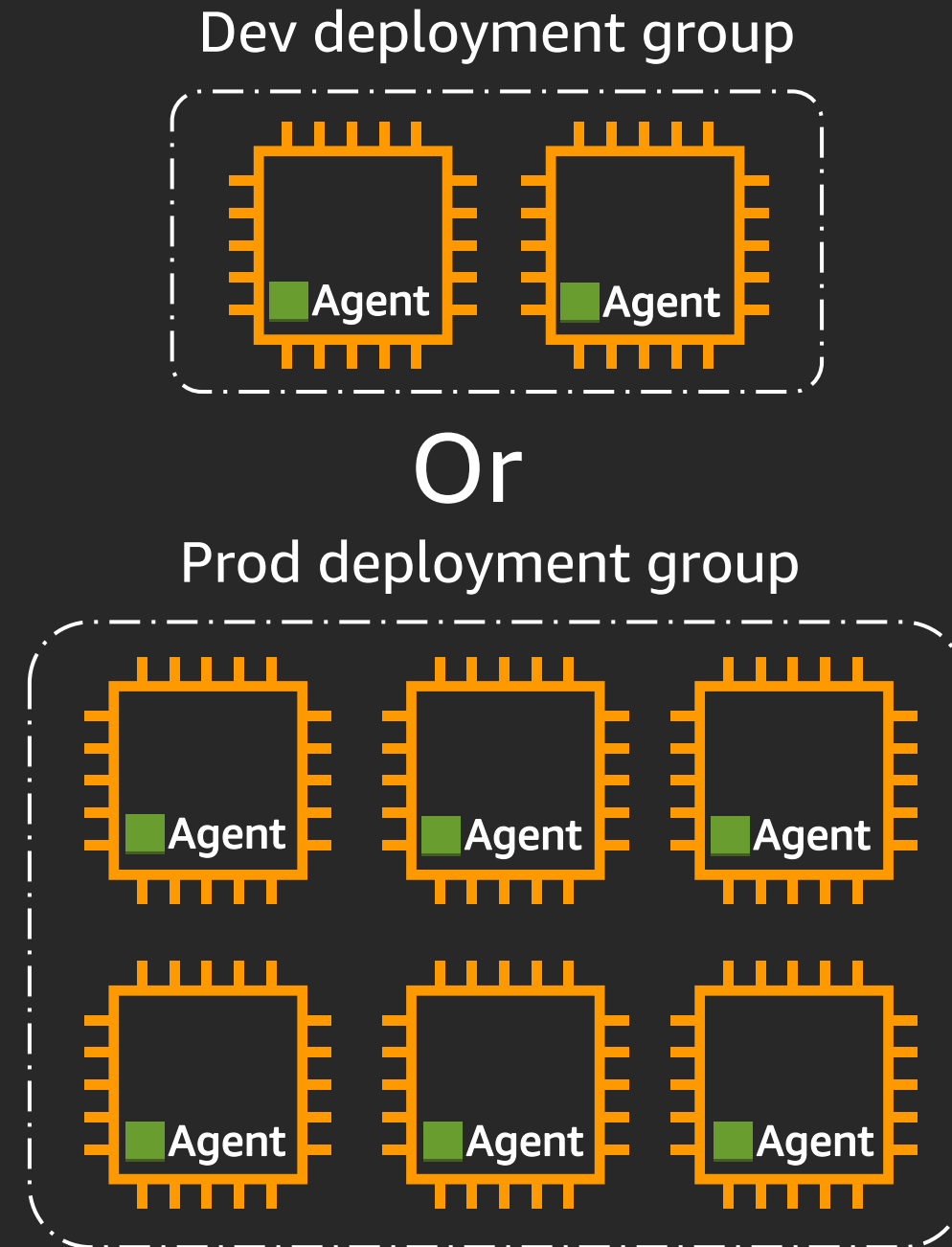
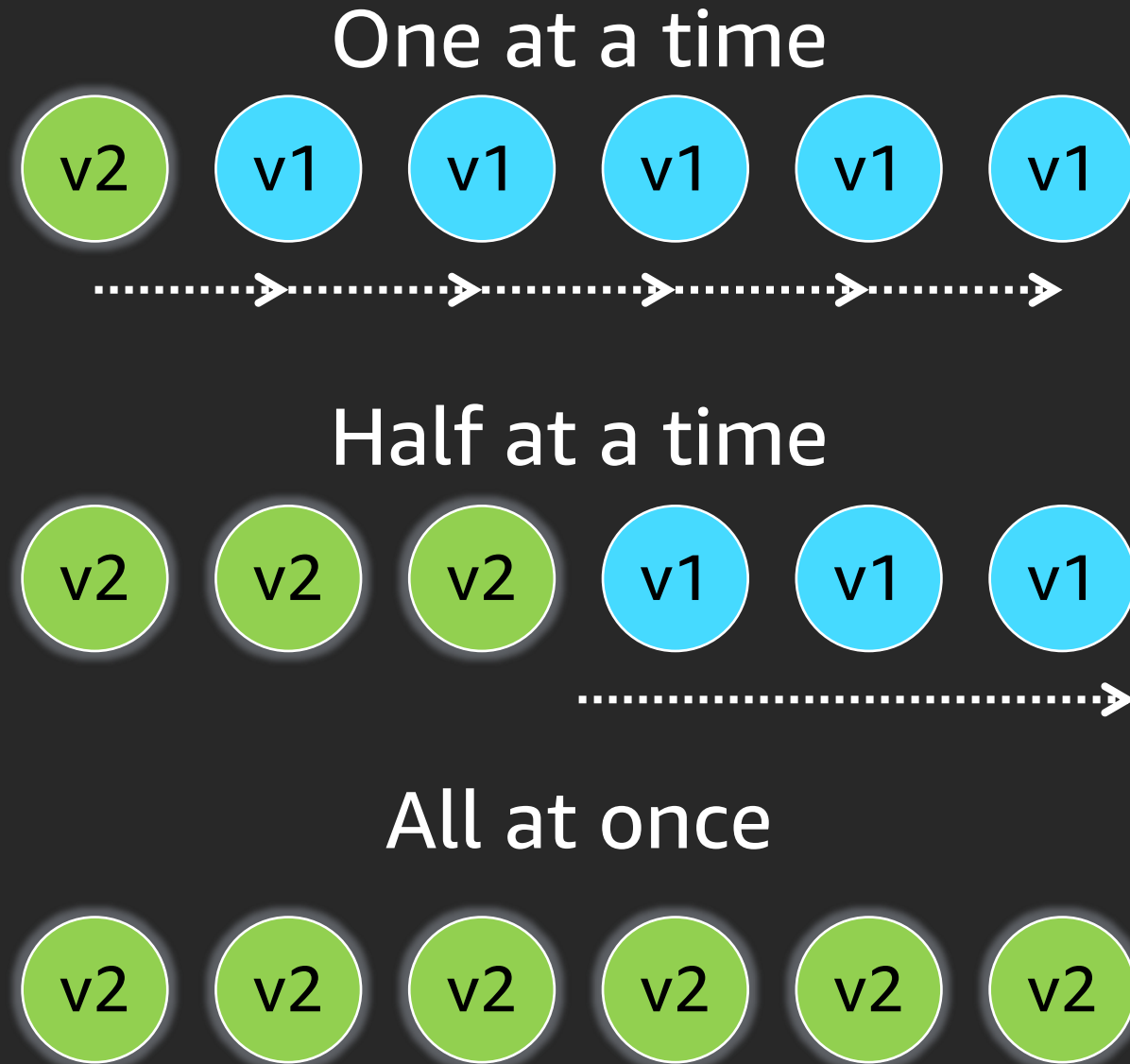
```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
permissions:
  - object: /var/www/html
    pattern: "*.html"
    owner: root
    group: root
    mode: 755
hooks:
  ApplicationStop:
    - location: scripts/deregister_from_elb.sh
  BeforeInstall:
    - location: scripts/install_dependencies.sh
  ApplicationStart:
    - location: scripts/start_httpd.sh
  validateService:
    - location: scripts/test_site.sh
    - location: scripts/register_with_elb.sh
```

- Send application files to one directory and configuration files to another

- Set specific permissions on specific directories & files

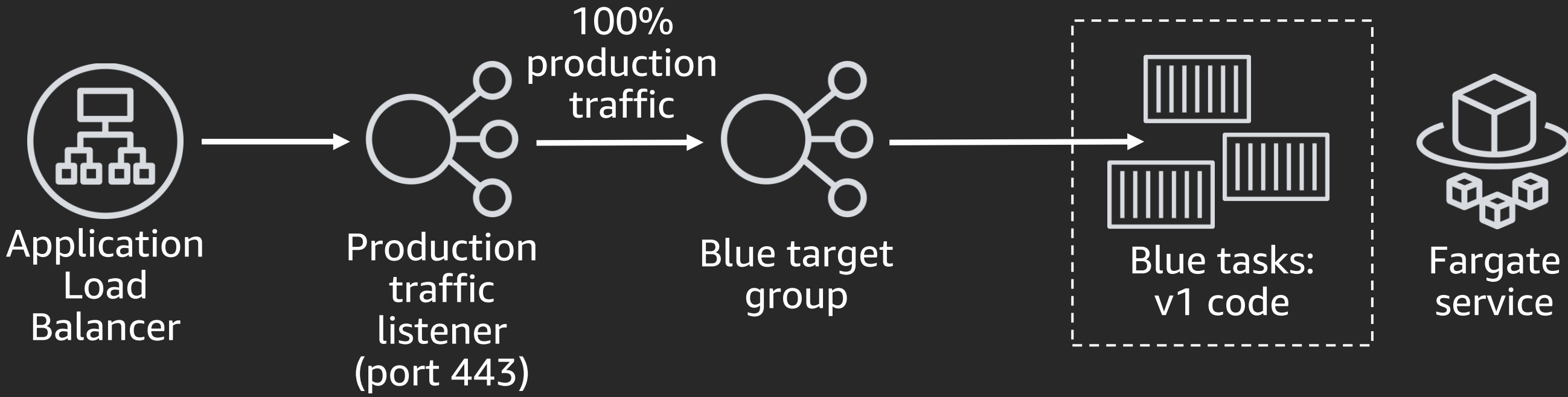
- Remove/add instance to Elastic Load Balancing
- Install dependency packages
- Start web server
- Confirm successful deploy

# Choose deployment speed and group

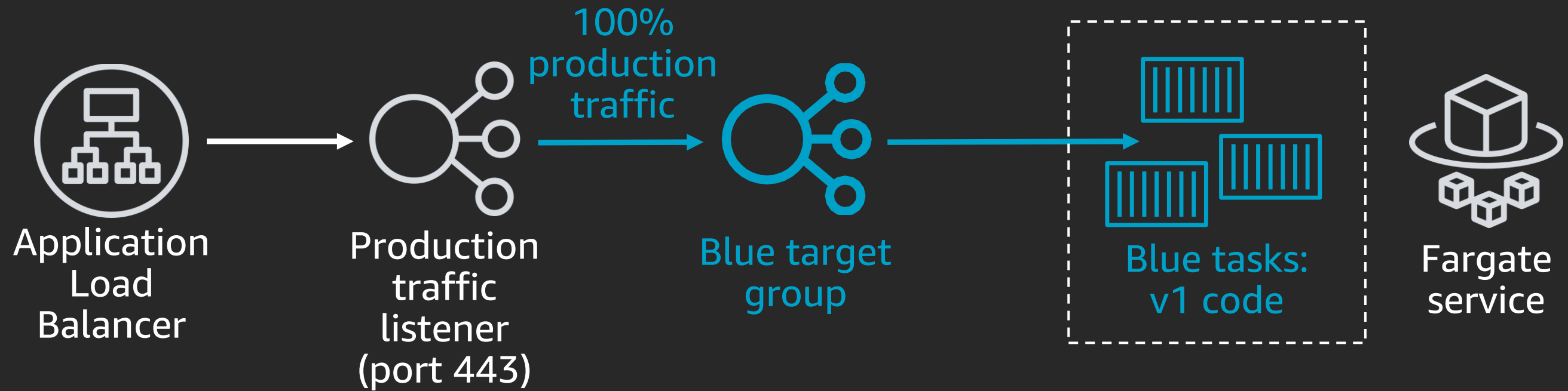




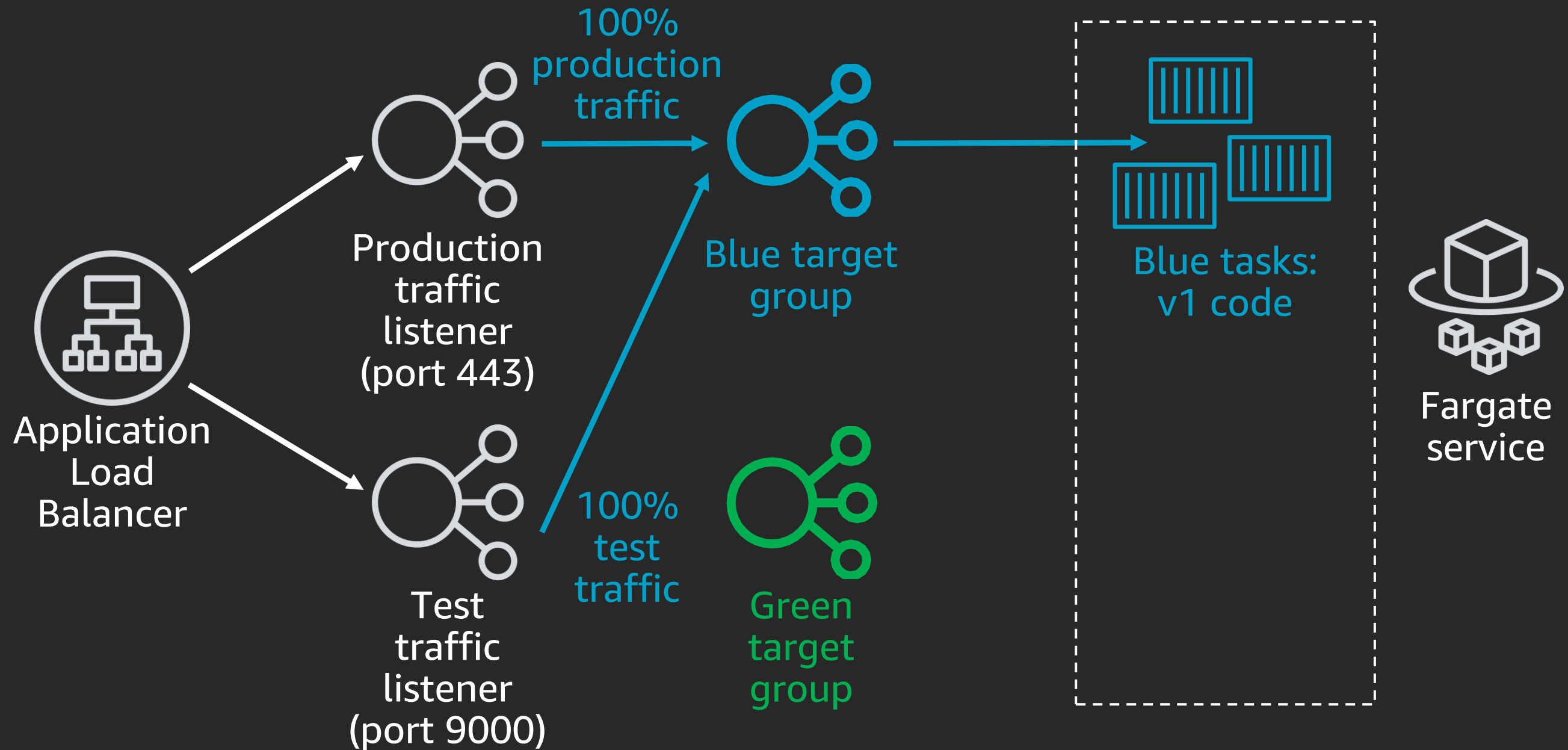
# CodeDeploy: Amazon ECS blue/green deployment



# CodeDeploy: Amazon ECS blue/green deployment

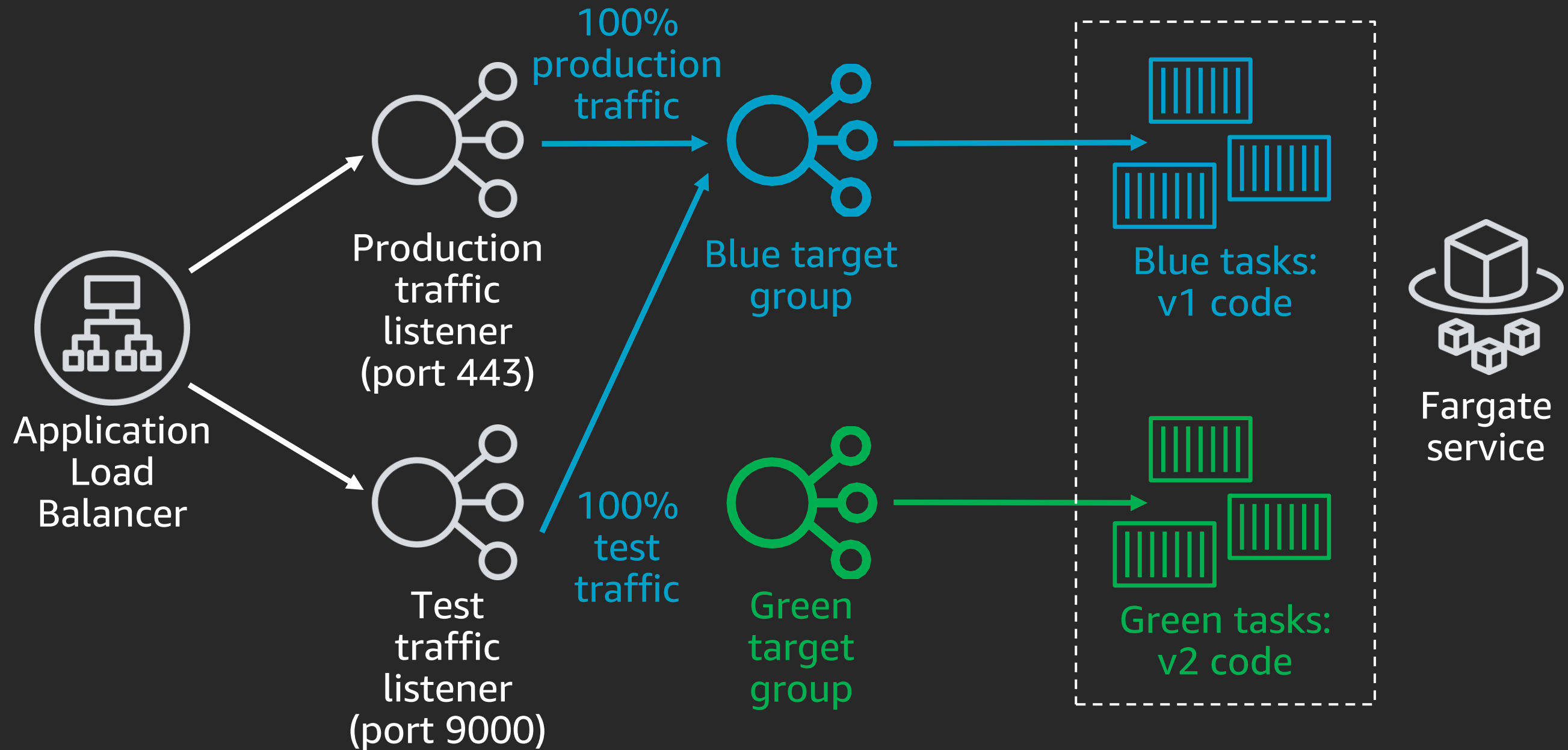


# CodeDeploy: Amazon ECS blue/green deployment



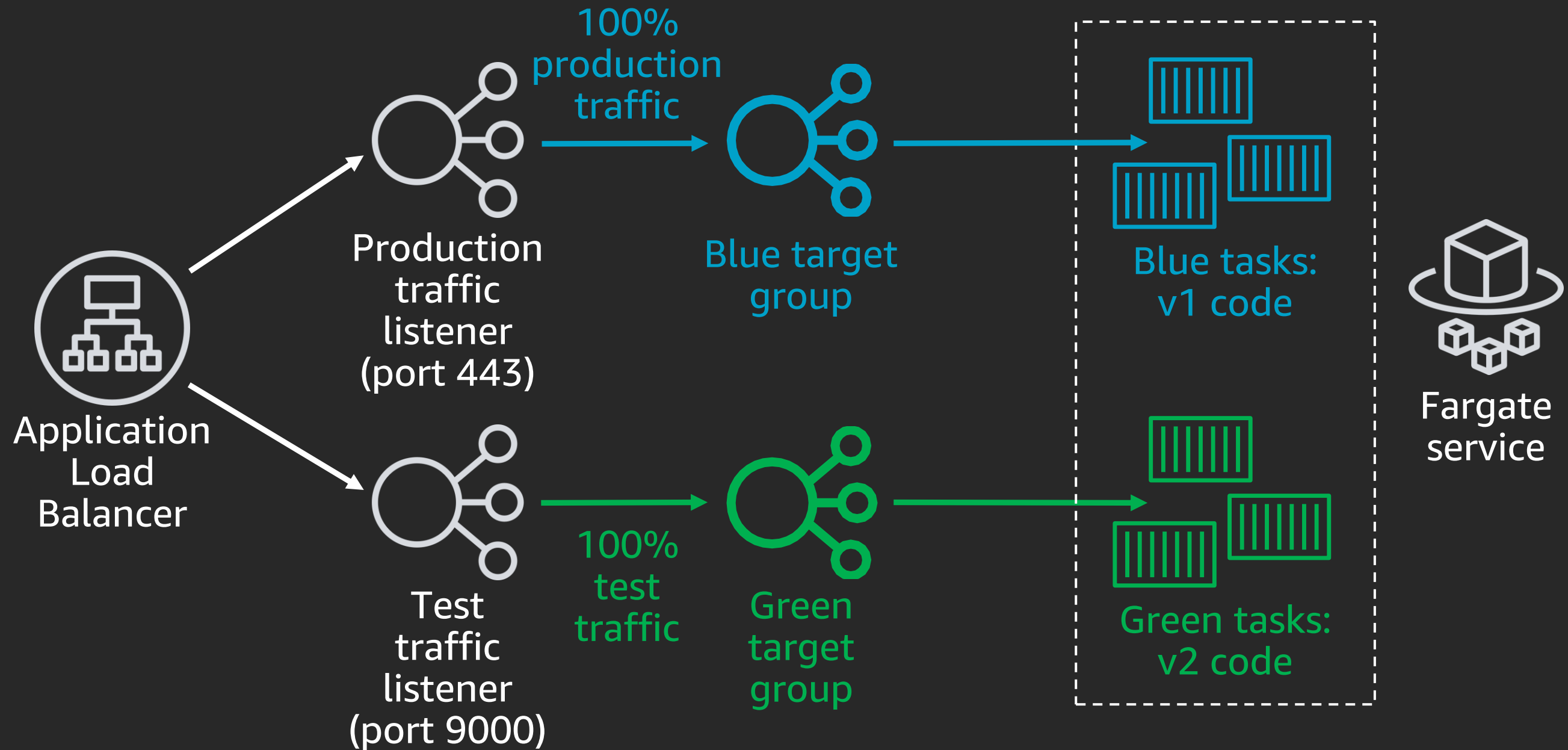
# CodeDeploy: Amazon ECS blue/green deployment

## Provision green tasks



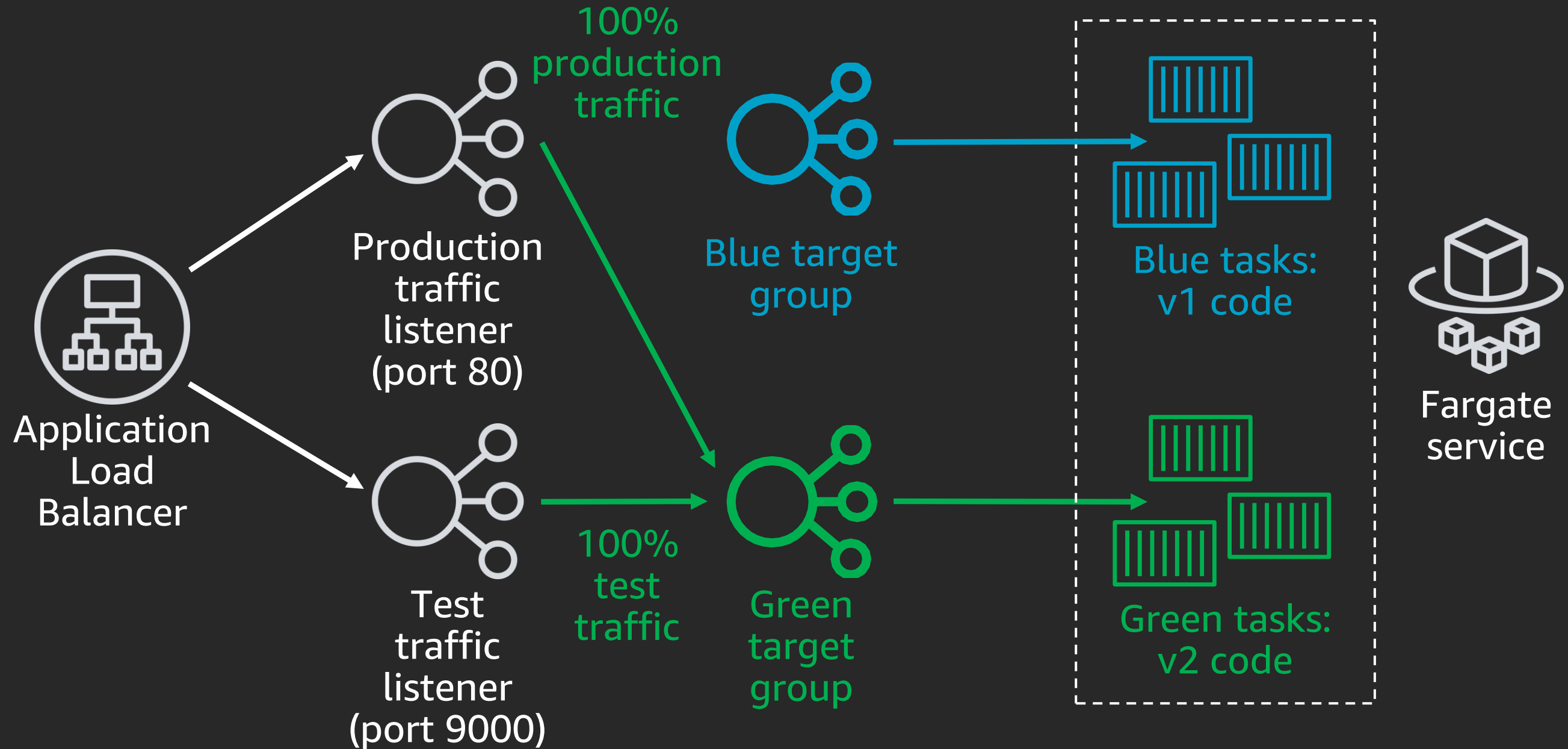
# CodeDeploy: Amazon ECS blue/green deployment

Shift test traffic to green; run validation tests against test endpoint



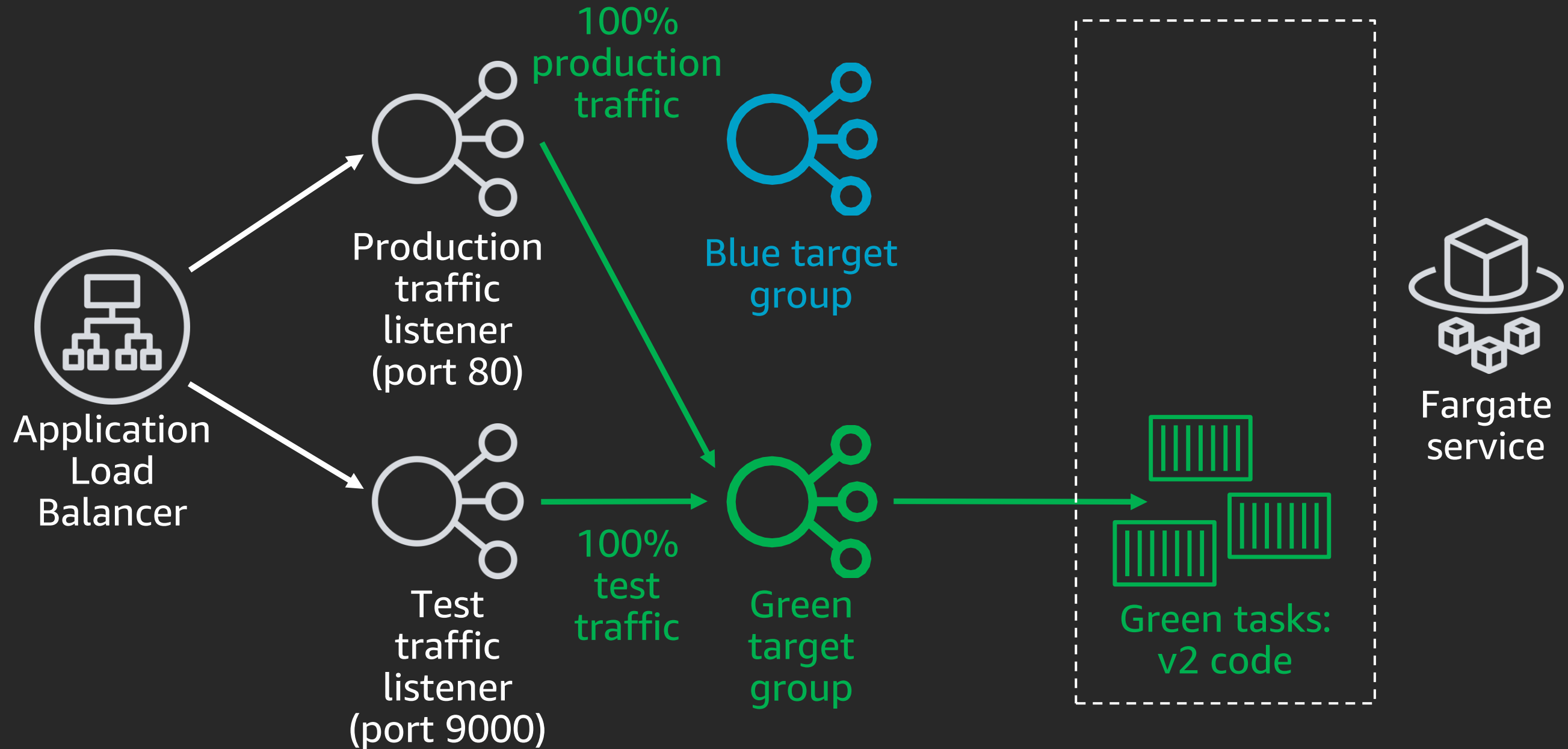
# CodeDeploy: Amazon ECS blue/green deployment

Shift production traffic to green; roll back in case of alarm



# CodeDeploy: Amazon ECS blue/green deployment

## Drain blue tasks

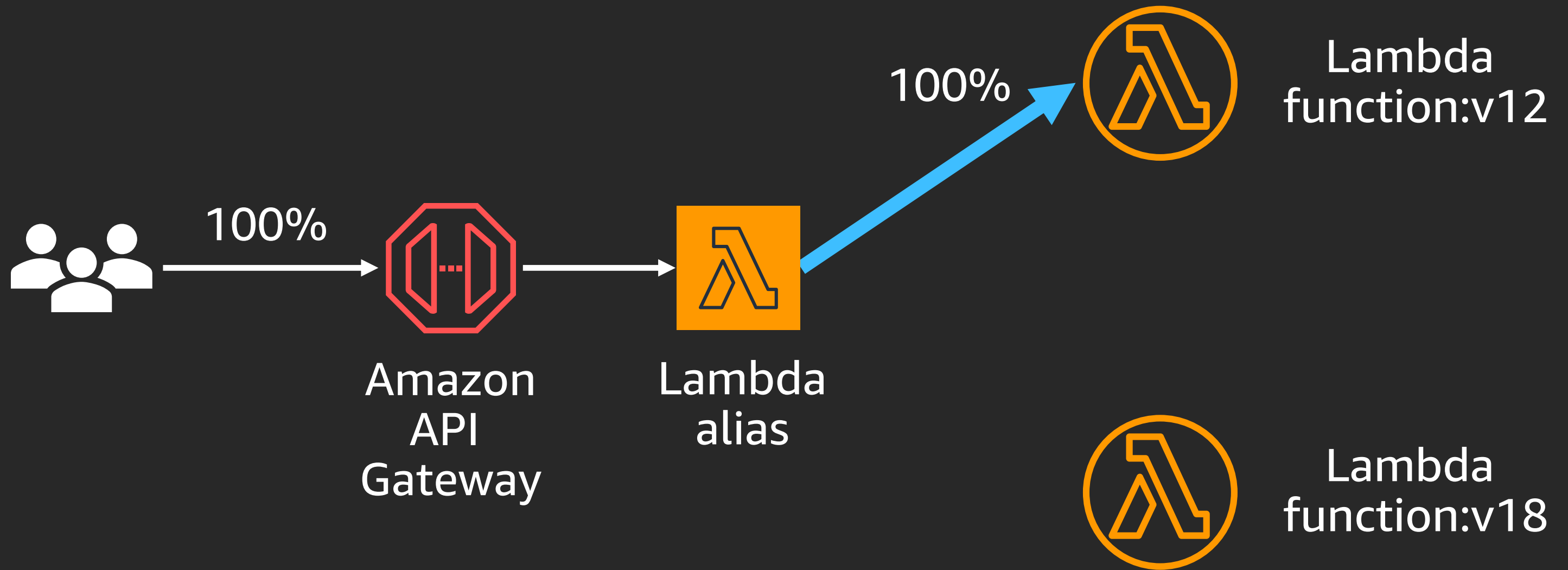


# AWS CodeDeploy: AWS Lambda deployments

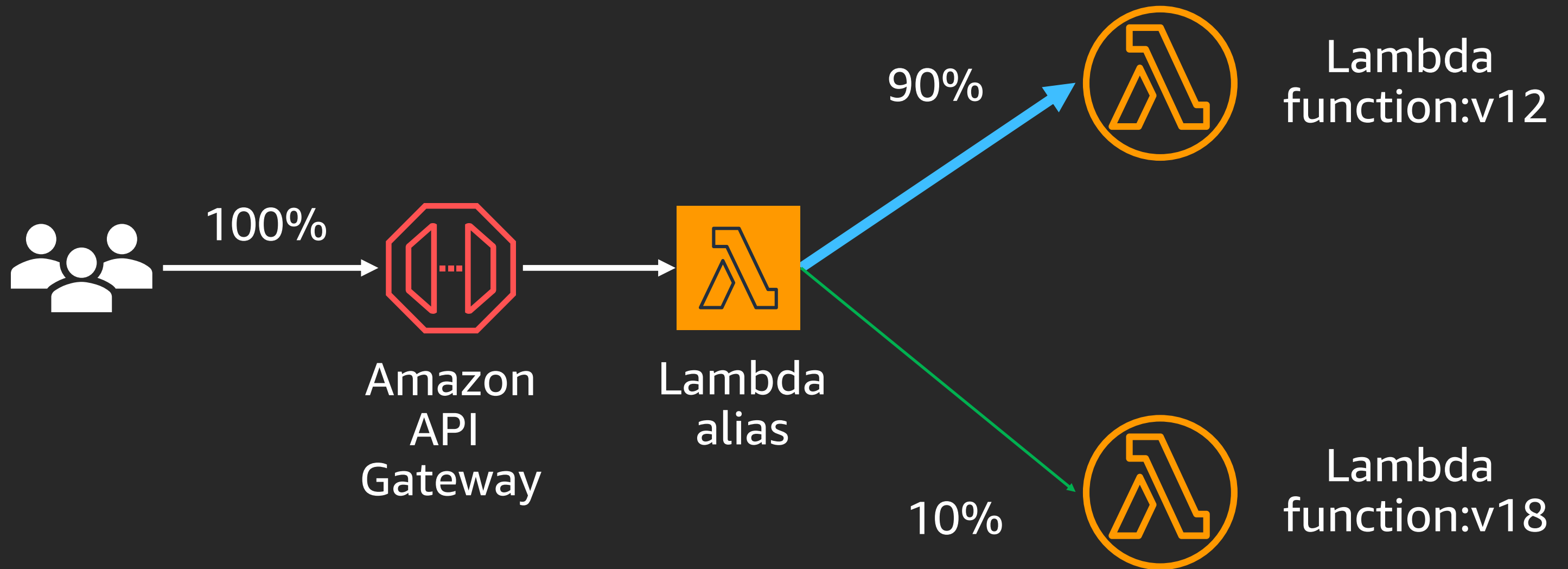




# AWS CodeDeploy: AWS Lambda deployments

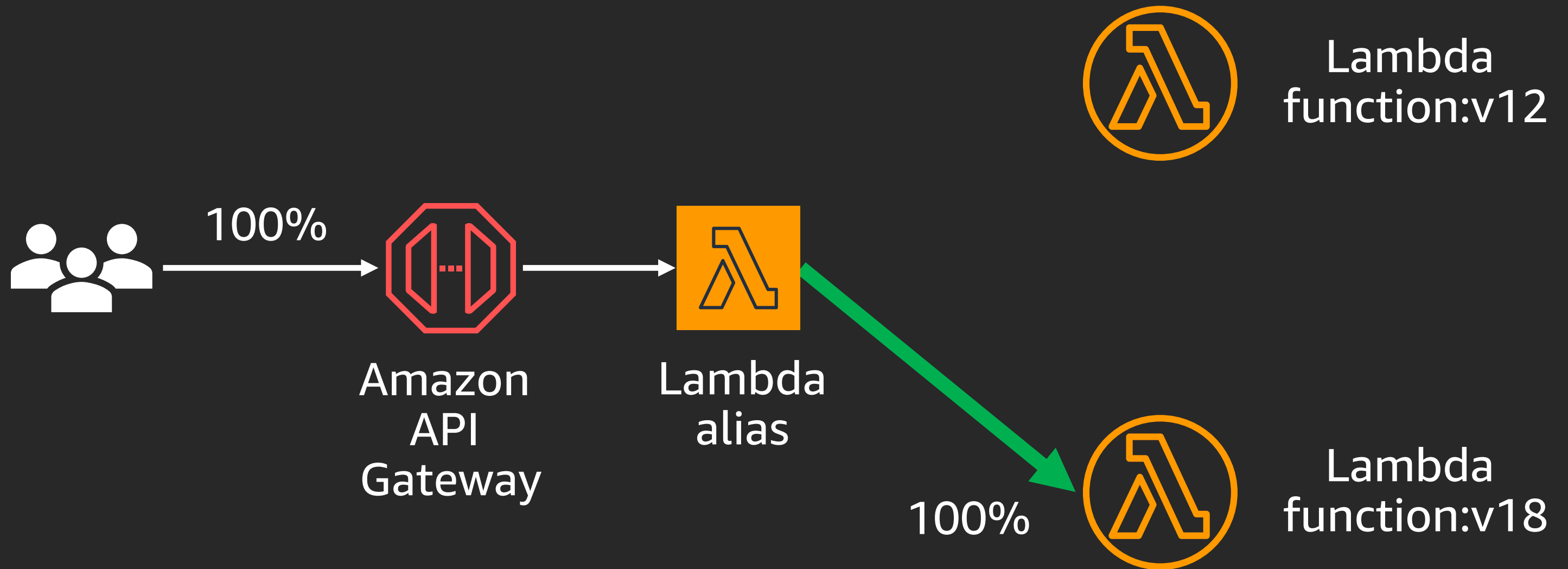


# AWS CodeDeploy: AWS Lambda deployments



Canary: "shift 10% of traffic for 10 mins., then shift the rest"

# AWS CodeDeploy: AWS Lambda deployments



Canary: "shift 10% of traffic for 10 mins., then shift the rest"

# Best practices for CI/CD

1

Pipeline  
automation

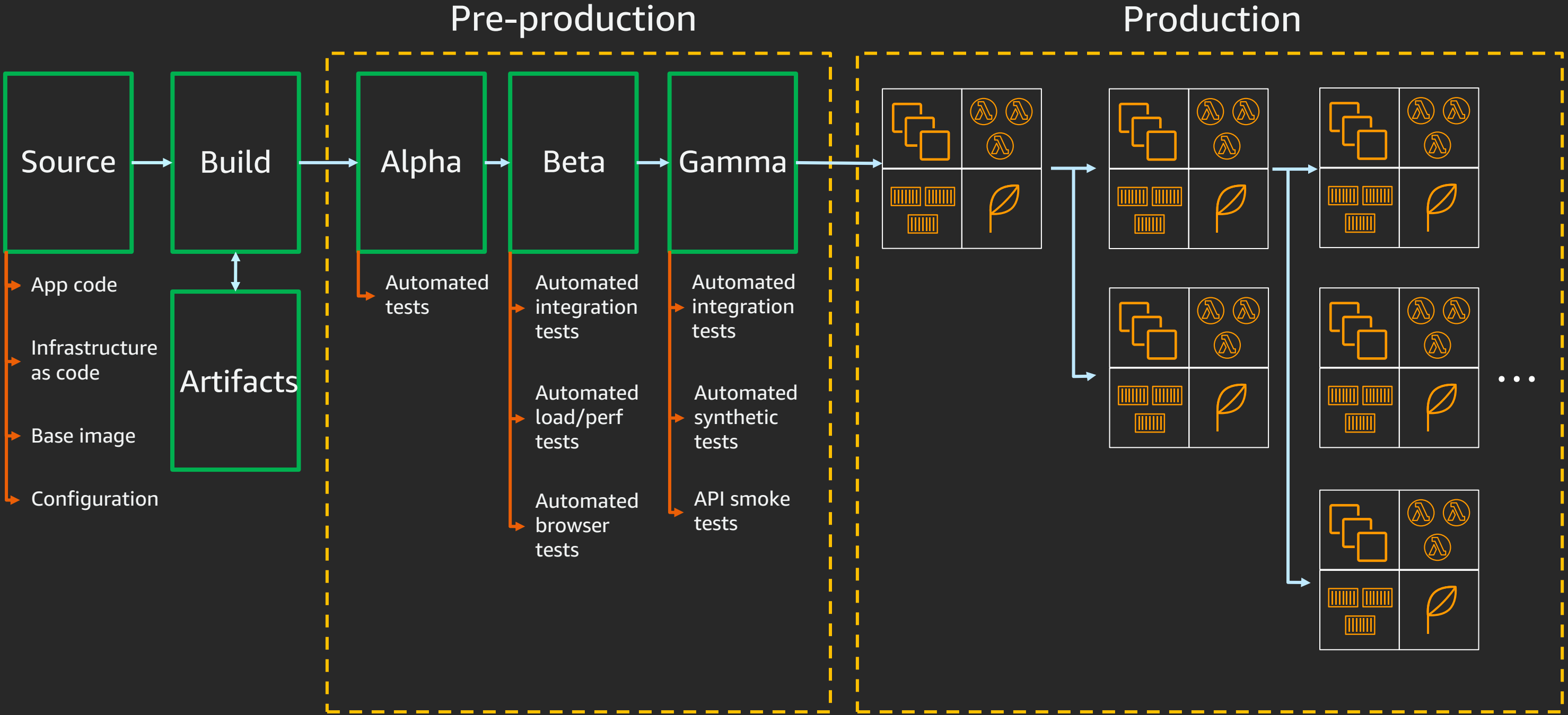
2

Safe  
deployments

3

Repeatable  
infrastructure  
changes

# What is DevOps at scale?



# Infrastructure as code goals



1. Make infrastructure changes repeatable and predictable
2. Release infrastructure changes using the same tools as code changes
3. Replicate production environment in a staging environment to enable continuous testing

# AWS Cloud Development Kit (AWS CDK)



- Open-source framework to define cloud infrastructure in Typescript, Python, Java & .NET
- Provisions resources with AWS CloudFormation
- Supports all AWS CloudFormation resource types
- Provides library of higher-level resource types that have AWS best practices built in by default

# AWS CDK template

```
import ec2 = require('@aws-cdk/aws-ec2');
import ecs = require('@aws-cdk/aws-ecs');
import cdk = require('@aws-cdk/cdk');

class BonjourFargate extends cdk.Stack {
  constructor(parent: cdk.App, name: string, props?: cdk.StackProps) {
    super(parent, name, props);

    const vpc = new ec2.VpcNetwork(this, 'MyVpc', { maxAZs: 2 });
    const cluster = new ecs.Cluster(this, 'Cluster', { vpc });

    new ecs.LoadBalancedFargateService(
      this, "FargateService", {
        cluster,
        image: ecs.DockerHub.image("amazon/amazon-ecs-sample"),
      });
  }
}

const app = new cdk.App();
new BonjourFargate(app, 'Bonjour');
app.run();
```

High-level virtual private cloud (VPC) class includes VPC, subnets, security groups, internet gateway, NAT gateways, and route tables





# AWS CDK template

```
import ec2 = require('@aws-cdk/aws-ec2');
import ecs = require('@aws-cdk/aws-ecs');
import cdk = require('@aws-cdk/cdk');

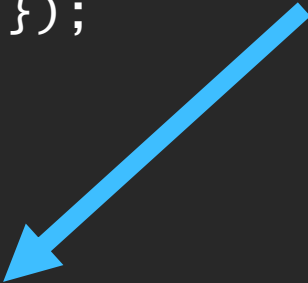
class BonjourFargate extends cdk.Stack {
  constructor(parent: cdk.App, name: string, props?: cdk.StackProps) {
    super(parent, name, props);

    const vpc = new ec2.VpcNetwork(this, 'MyVpc', { maxAZs: 2 });
    const cluster = new ecs.Cluster(this, 'Cluster', { vpc });

    new ecs.LoadBalancedFargateService(
      this, "FargateService", {
        cluster,
        image: ecs.DockerHub.image("amazon/amazon-ecs-sample"),
      });
  }
}

const app = new cdk.App();
new BonjourFargate(app, 'Bonjour');
app.run();
```

High-level Fargate class includes Amazon ECS service, Amazon ECS task definition, Application Load Balancer, listener rule, target group, and, optionally, Amazon Route 53 alias record



# AWS CDK template

```
import ec2 = require('@aws-cdk/aws-ec2');
import ecs = require('@aws-cdk/aws-ecs');
import cdk = require('@aws-cdk/cdk');

class BonjourFargate extends cdk.Stack {
  constructor(parent: cdk.App, name: string, props?: cdk.StackProps) {
    super(parent, name, props);

    const vpc = new ec2.VpcNetwork(this, 'MyVpc', { maxAZs: 2 });
    const cluster = new ecs.Cluster(this, 'Cluster', { vpc });

    new ecs.LoadBalancedFargateService(
      this, "FargateService", {
        cluster,
        image: ecs.DockerHub.image("amazon/amazon-ecs-sample"),
      });
  }
}

const app = new cdk.App();
new BonjourFargate(app, 'Bonjour');
app.run();
```

22 lines of  
TypeScript code  
generate over  
400 lines of AWS  
CloudFormation  
syntax

# CI/CD @ Electrify Asia

# Electrify Asia

- Energy technology company
- Build sustainable energy ecosystems through development of transactive energy platforms
- Democratized access to clean energy across Asia-Pacific



# Challenges we had

- Lacking a standard CI/CD platform
- More manual human interact workload for deployments
- Highly vulnerable security issues and trouble with keeping the secrets
- Trouble managing the infrastructure
- Hard to isolate the bottlenecks of the application/services, so there is no proper observability

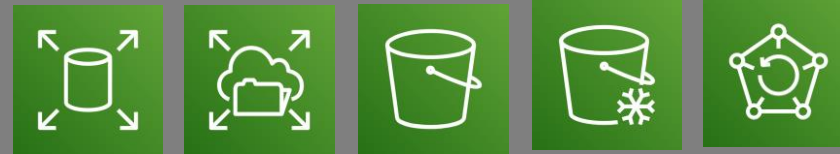
# Our AWS Stack

## Compute



Amazon EC2   Amazon EKS   Amazon ECR   AWS Lambda

## Storage



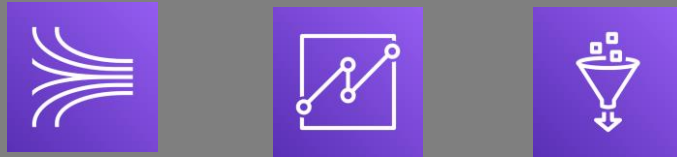
Amazon EBS   Amazon EFS   Amazon S3   Amazon S3 Glacier   AWS Backup

## Database



Amazon Aurora   Amazon ElastiCache   Amazon Redshift   Amazon DynamoDB

## Analytics



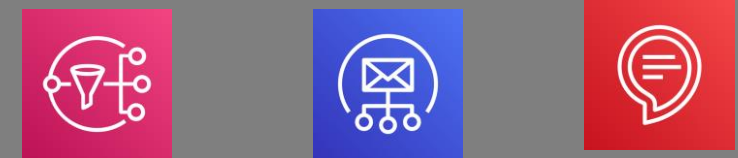
Amazon Kinesis   Amazon QuickSight   AWS Glue

## Developer tools



AWS CodePipeline   AWS CodeBuild   AWS CodeCommit   AWS CodeDeploy

## Customer engagement and other



Amazon SNS   Amazon SES   Amazon Alexa

## Security, identity & compliance



AWS Certificate Manager   AWS WAF   IAM   AWS Secrets Manager   AWS KMS

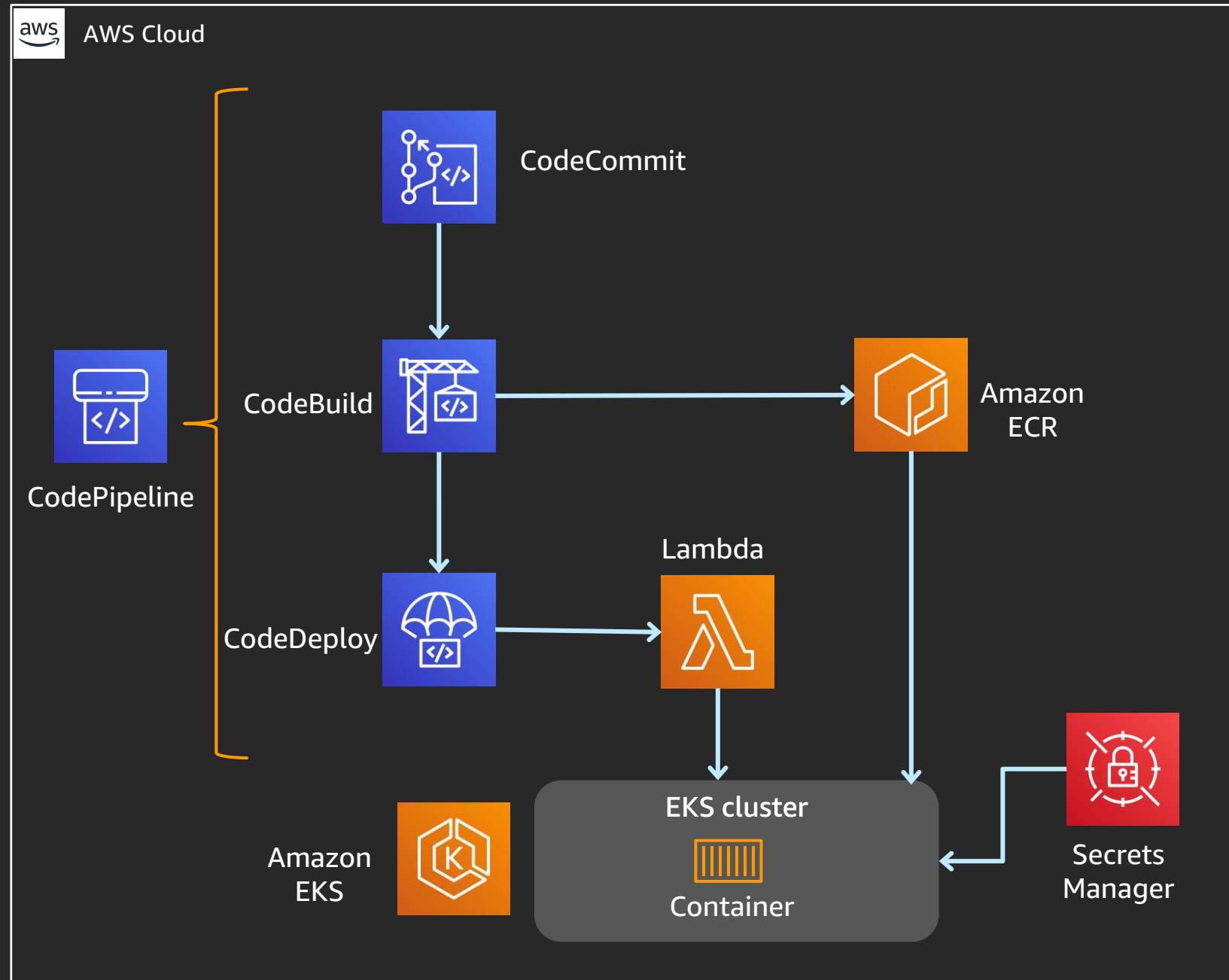
## Management & governance



AWS Trusted Advisor   AWS CloudTrail   AWS Organizations   AWS CloudFormation   AWS Systems Manager   Amazon CloudWatch   AWS Auto Scaling

“We know that we have to deploy things faster and break things over and over again. To make that process streamlined, we came up with this solution.”

# DevOps pipeline

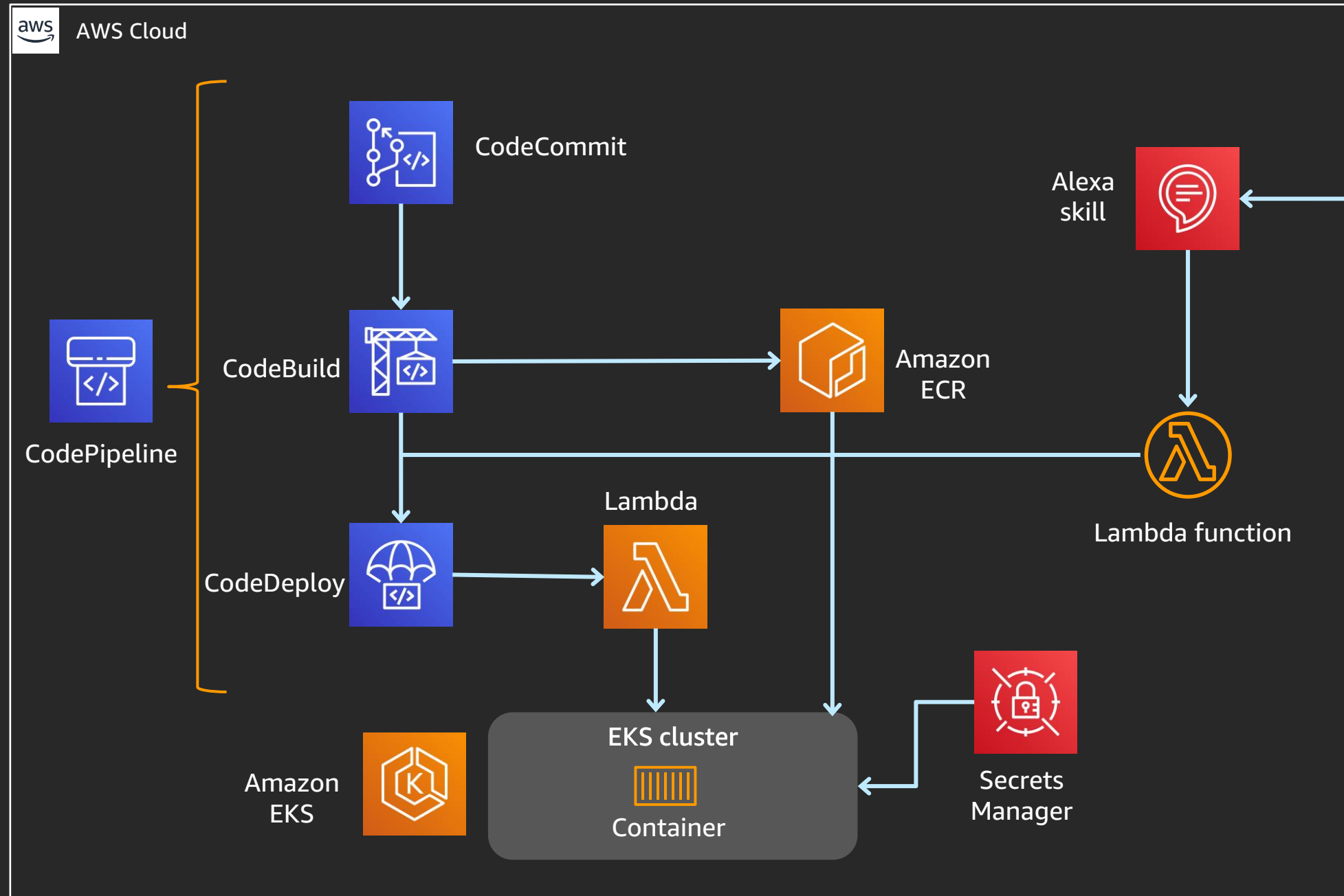


- Saved time and costs
- Everything is automated
- Used Secrets Manager to store the secure configs
- Container Insights and CloudWatch provided observability



“... and we made it more interesting by integrating Alexa with AWS CodePipeline.”

# DevOps pipeline v2



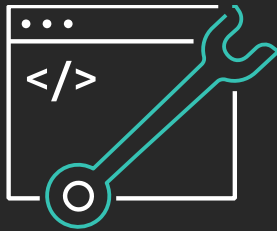
amazon alexa

Alexa to trigger the deployments quickly and easily

# Demo

# Learn to build modern applications on AWS

Resources created by the experts at AWS to help you build and validate developer skills



Enable rapid innovation by developing your skills in designing, building, and managing modern applications



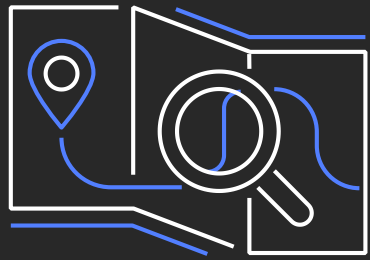
Learn to modernize your applications with free digital training and classroom offerings, including Architecting on AWS, Developing on AWS, and DevOps Engineering on AWS



Validate expertise with the AWS Certified DevOps – Professional or AWS Certified Developer – Associate exams

Visit the developer learning path at [aws.amazon.com/training/path-developing](https://aws.amazon.com/training/path-developing)

# AWS Training and Certification



## Training for the whole team

Explore tailored learning paths for customers and partners



## Flexibility to learn your way

Build cloud skills with 550+ free digital training courses, or dive deep with classroom training



## Validate skills with AWS Certification

Demonstrate expertise with an industry-recognized credential



## Educational programs

Find entry-level cloud talent with AWS Academy and AWS re/Start

[aws.amazon.com/training](https://aws.amazon.com/training)

# Thank you!

Loh Yiang Meng  
ymloh@amazon.com