aws modern apps

CI/CD for Containers: A way forward for your DevOps Pipeline

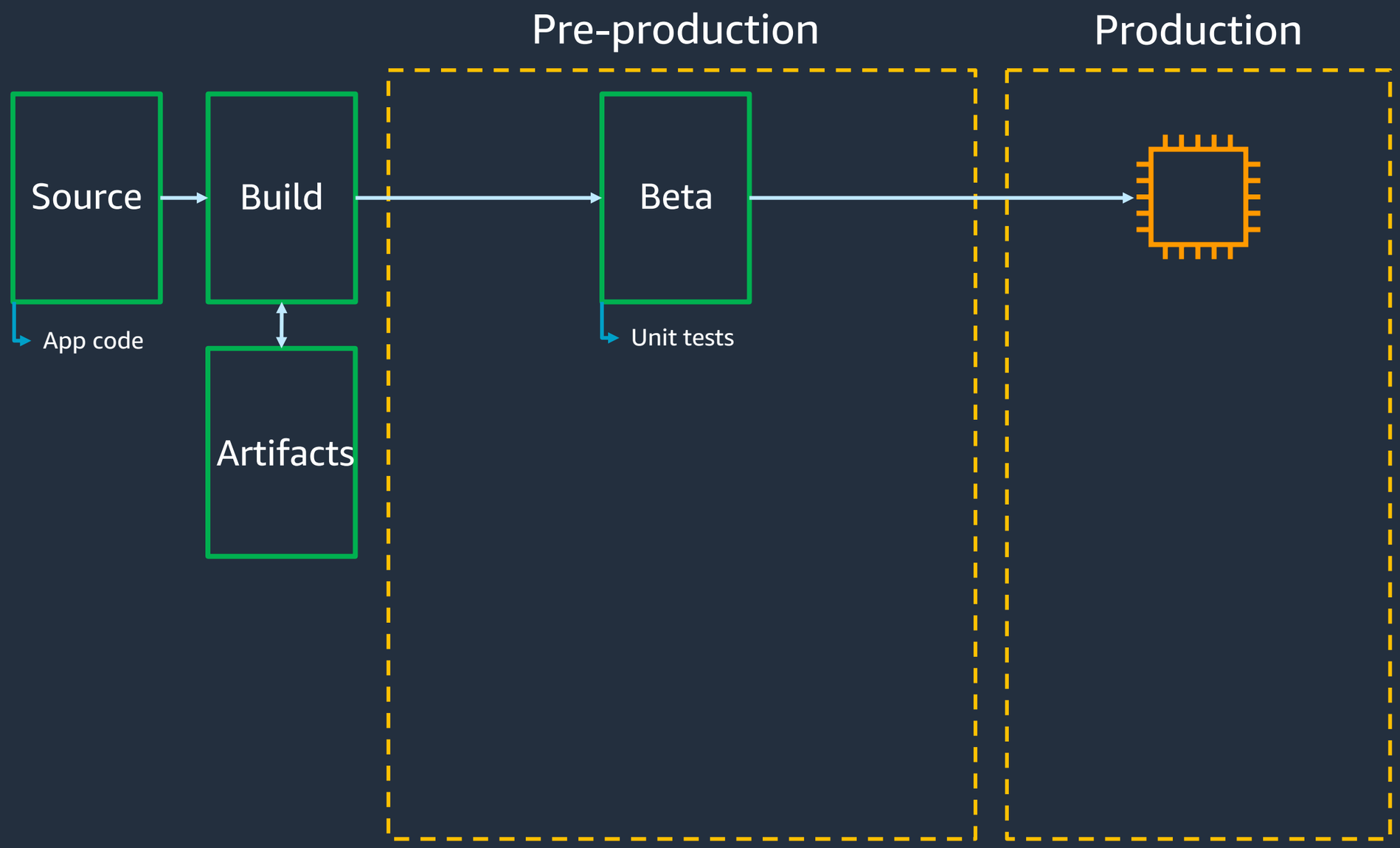
KJ Pittl
ISV Solutions Architect,
Amazon Web Services

Agenda

- ~~What is DevOps,~~ what are the challenges at scale?
- ~~modernise your application, build containers, container registry,~~
- ~~What's this ECS/EKS on EC2 or Fargate~~
- Pipeline Automation
- Safe Deployments
- Repeatable infrastructure changes

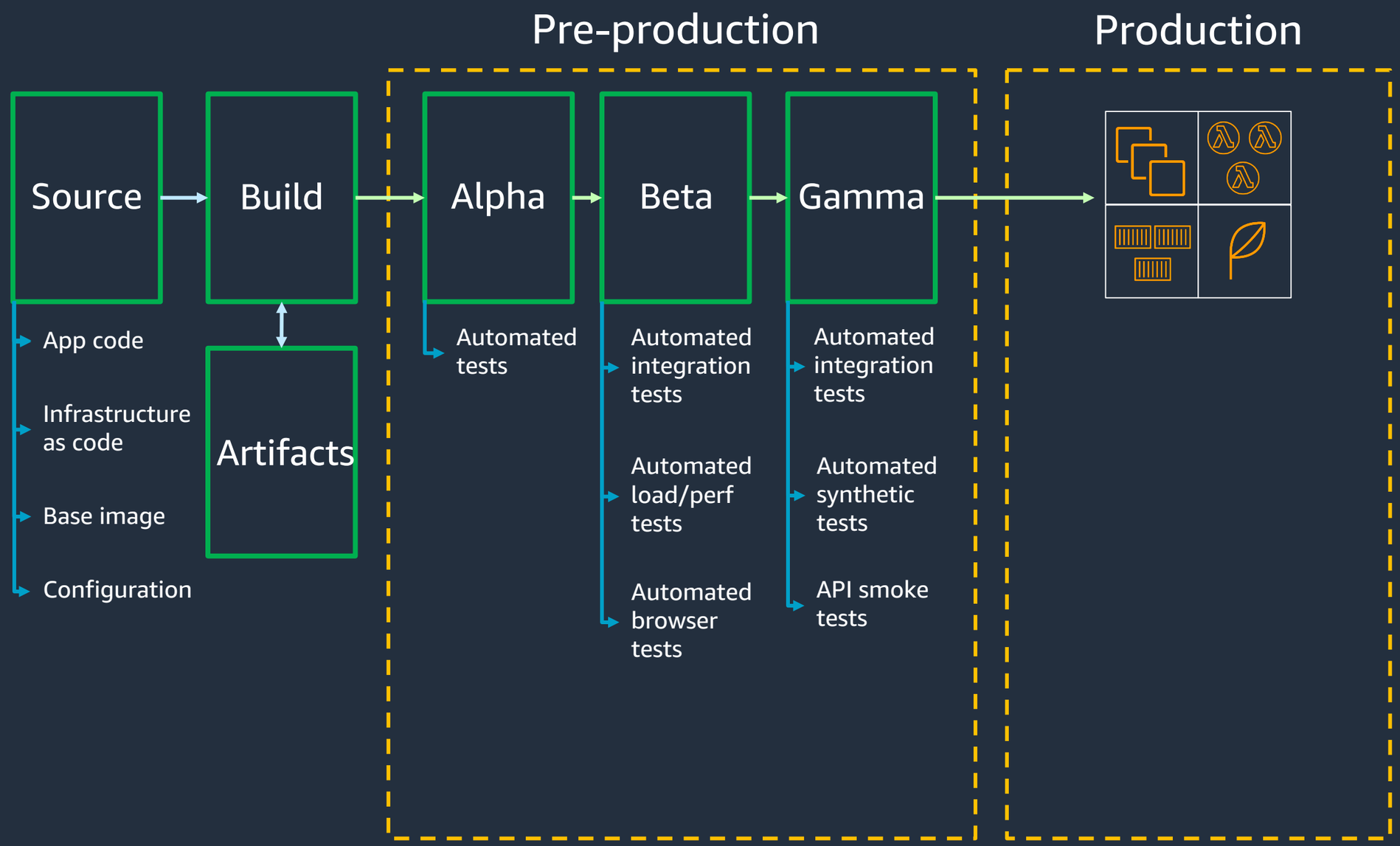
aws modern apps

What is DevOps at scale?



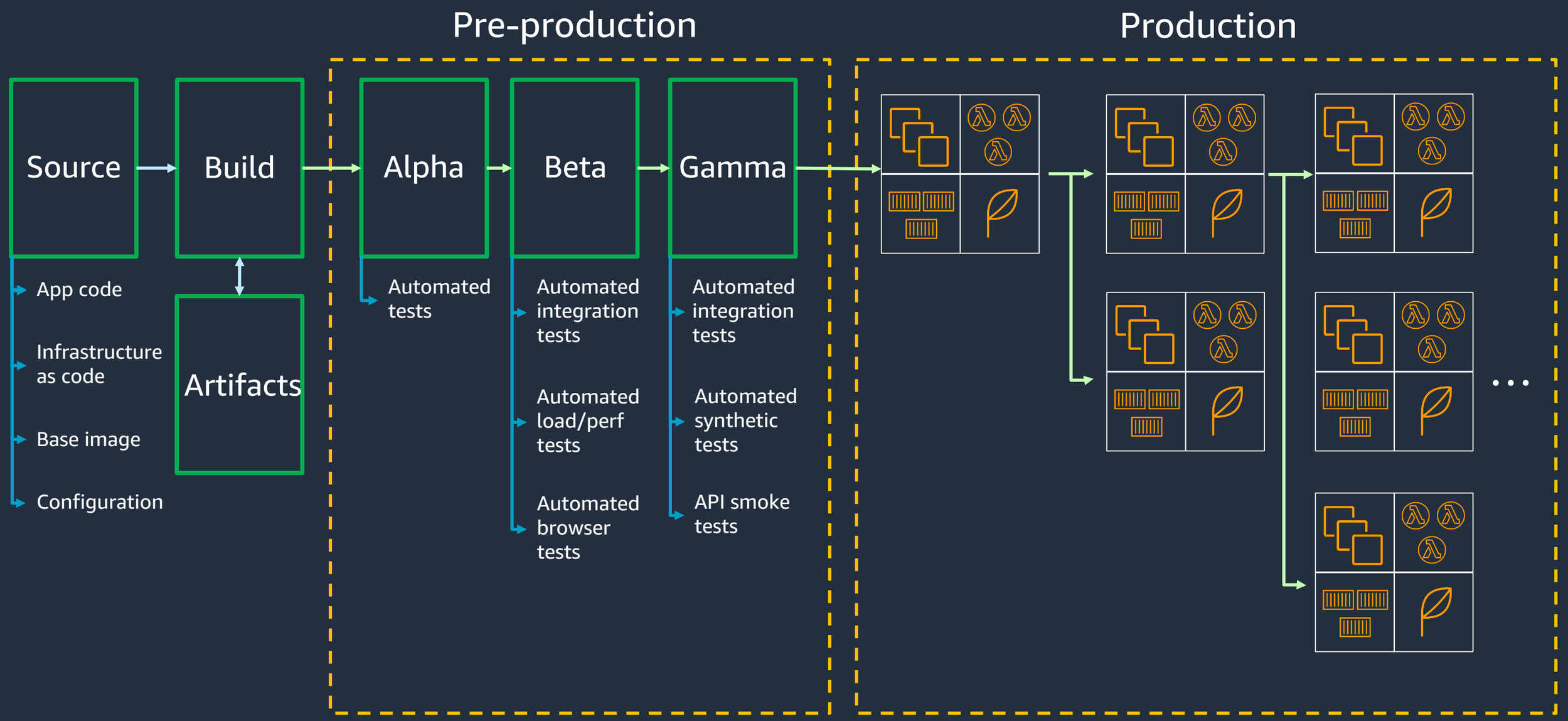
aws modern apps

What is DevOps at scale?



aws modern apps

What is DevOps at scale?



aws modern apps

Best practices for CI/CD

1

Pipeline Automation

2

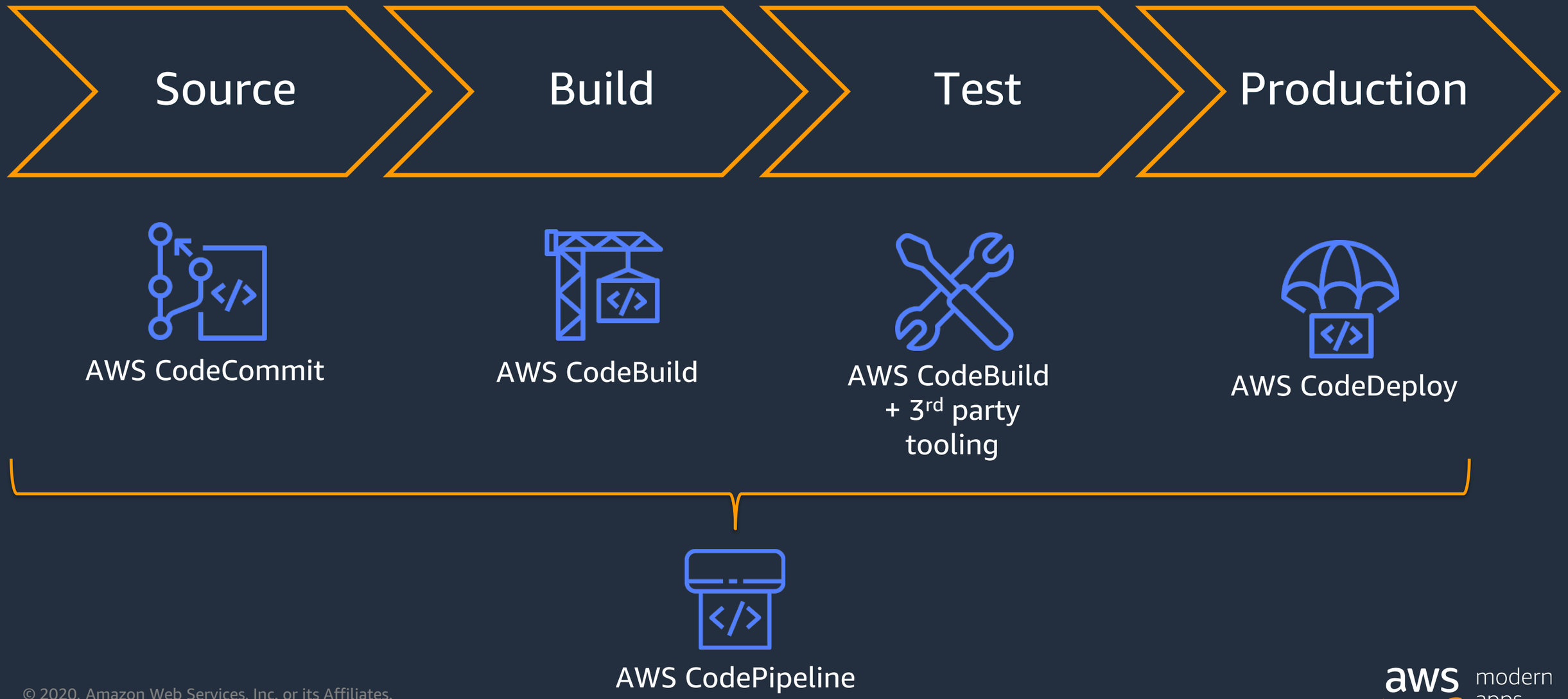
Safe Deployments

3

Repeatable infrastructure changes

aws modern apps

AWS Code services



aws modern apps

AWS CodePipeline

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

aws modern apps

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 modern apps

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 (SAM)

AWS Lambda

aws modern apps

AWS CodePipeline: Supported triggers

Automatically kick off release

Amazon EventBridge / Cloudwatch Events

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

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

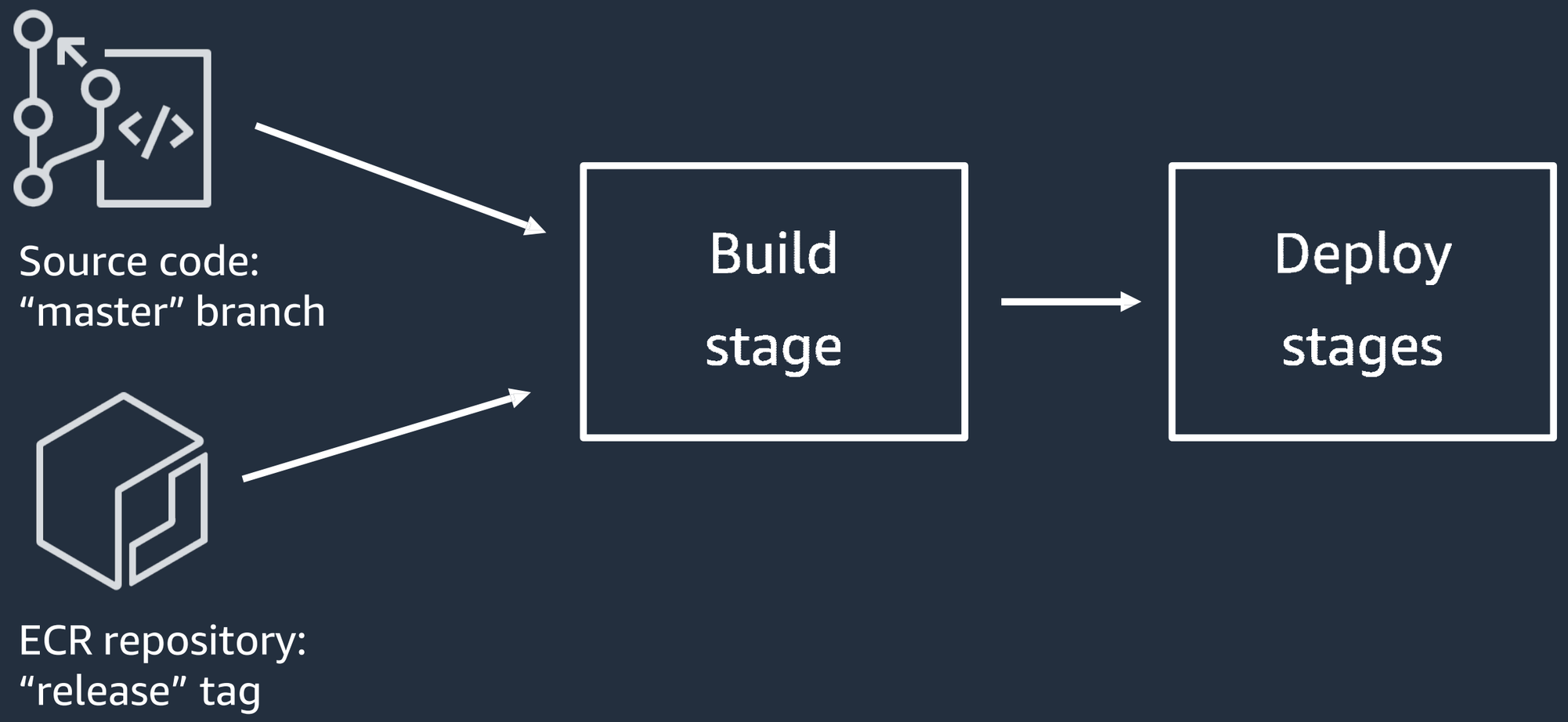
Webhooks

- Docker Hub
- Quay
- Artifactory

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

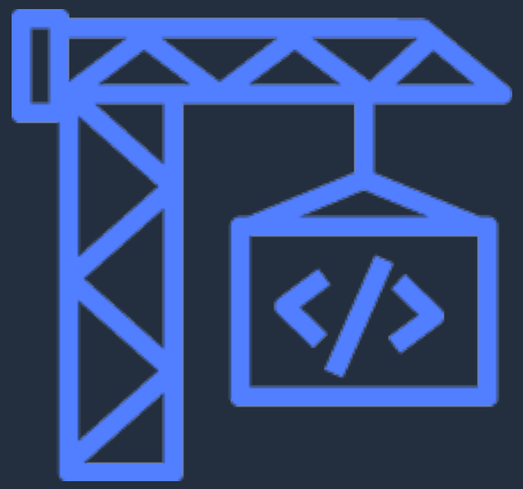
aws modern apps

AWS CodePipeline: ECR source action



aws modern apps

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: Docker buildspec

```
version: 0.2

phases:
  build:
    commands:
      - $(aws ecr get-login --no-include-email)
      - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG .
      - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG $ECR_REPO:$IMAGE_TAG
      - docker push $ECR_REPO:$IMAGE_TAG
```


aws modern apps

AWS CodeBuild: EKS buildspec

```
version: 0.2
phases:
  install:
    commands:
      - curl -sS -o aws-iam-authenticator https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/aws-iam-authenticator
      - curl -sS -o kubectl https://amazon-eks.s3-us-west-2.amazonaws.com/1.10.3/2018-07-26/bin/linux/amd64/kubectl
      - chmod +x ./kubectl ./aws-iam-authenticator
      - export PATH=$PWD/:$PATH
  -   apt-get update && apt-get -y install jq python3-pip python3-dev && pip3 install --upgrade
  awscli  pre_build:
      commands:
        - TAG="$REPOSITORY_NAME.$REPOSITORY_BRANCH.$ENVIRONMENT_NAME.$(date +%Y-%m-%d.%H.%M.%S).$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION |
          head -c  8)"
        - sed -i 's@CONTAINER_IMAGE@'"$REPOSITORY_URI:$TAG"'@' hello-k8s.yml
        - $(aws ecr get-login --no-include-email)
  -     export KUBECONFIG=$HOME/.kube/config
  build:
    commands:
      - docker build --tag $REPOSITORY_URI:$TAG .

  post_build:
    commands:
      - docker push $REPOSITORY_URI:$TAG
      - CREDENTIALS=$(aws sts assume-role --role-arn $EKS_KUBECTL_ROLE_ARN --role-session-name codebuild-kubectl --duration-seconds 900)
      - export AWS_ACCESS_KEY_ID="$(echo ${CREDENTIALS} | jq -r '.Credentials.AccessKeyId')"
      - export AWS_SECRET_ACCESS_KEY="$(echo ${CREDENTIALS} | jq -r '.Credentials.SecretAccessKey')"
      - export AWS_SESSION_TOKEN="$(echo ${CREDENTIALS} | jq -r '.Credentials.SessionToken')"
      - export AWS_EXPIRATION=$(echo ${CREDENTIALS} | jq -r '.Credentials.Expiration')
      - aws eks update-kubeconfig --name $EKS_CLUSTER_NAME
      - kubectl apply -f hello-k8s.yml
-     printf '[{"name":"hello-k8s","imageUri":"%s"}]' $REPOSITORY_URI:$TAG >
build.json  artifacts:
  files: build.json
```


aws modern apps

AWS CodeBuild: Lambda buildspec

```
version: 0.2
phases:
  build:
    commands:
      - npm ci
      - npm test
      - >
          aws cloudformation package
            --template-file template.yml
             --output-template template-output.yml
             --s3_bucket $BUCKET
artifacts:
  type: zip
  files:
    - template-output.yml
```


aws modern apps

Best practices for CI/CD

1

Pipeline automation

2

Safe deployments

3

Repeatable infrastructure changes

aws modern apps

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 modern apps

AWS CodeDeploy: Amazon EC2 deployments

```yaml
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

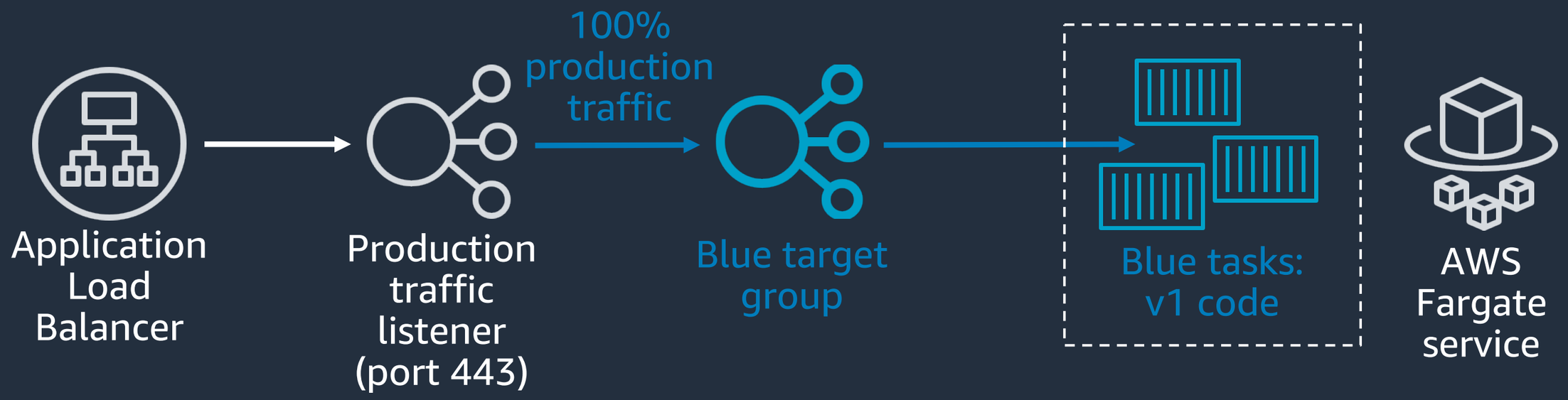
aws modern apps

CodeDeploy-ECS appspec

```
version: 1.0
Resources:
    - TargetService:
        Type: AWS::ECS::Service
        Properties:
            - TaskDefinition: "my_task_definition:8"
              LoadBalancerInfos:
                    - ContainerName: "SampleApp"
                      ContainerPort: 80
Hooks:
    - BeforeInstall: "LambdaFunctionToExecuteAnythingBeforeNewRevisionInstalltion"
    - AfterInstall: "LambdaFunctionToExecuteAnythingAfterNewRevisionInstallation"
    - AfterAllowTestTraffic: "LambdaFunctionToValidateAfterTestTrafficShift"
    - BeforeAllowTraffic: "LambdaFunctionToValidateBeforeTrafficShift"
    - AfterAllowTraffic: "LambdaFunctionToValidateAfterTrafficShift"
```


aws modern apps

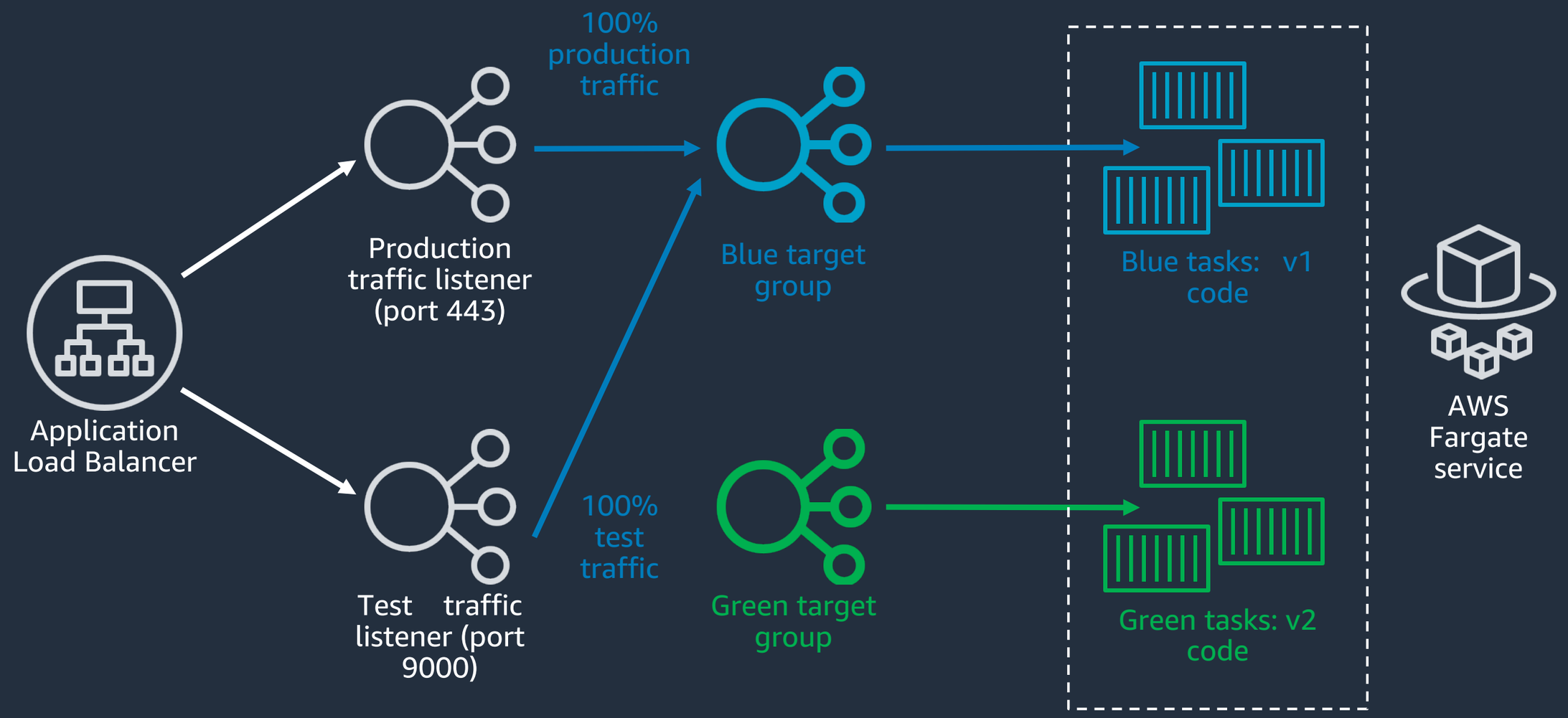
CodeDeploy: Amazon ECS blue/green deployment



aws modern apps

CodeDeploy: Amazon ECS blue/green deployment

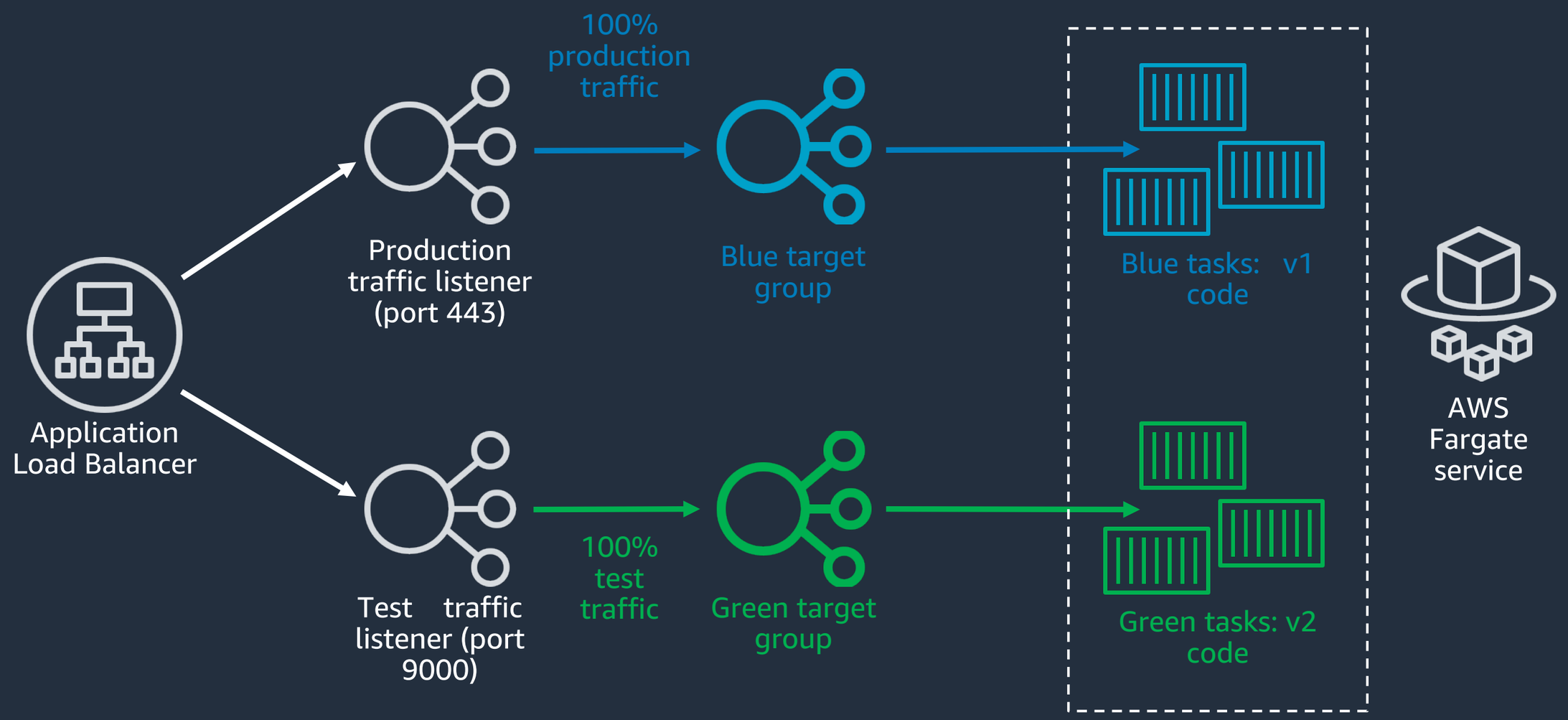
Provision green tasks



aws modern apps

CodeDeploy: Amazon ECS blue/green deployment

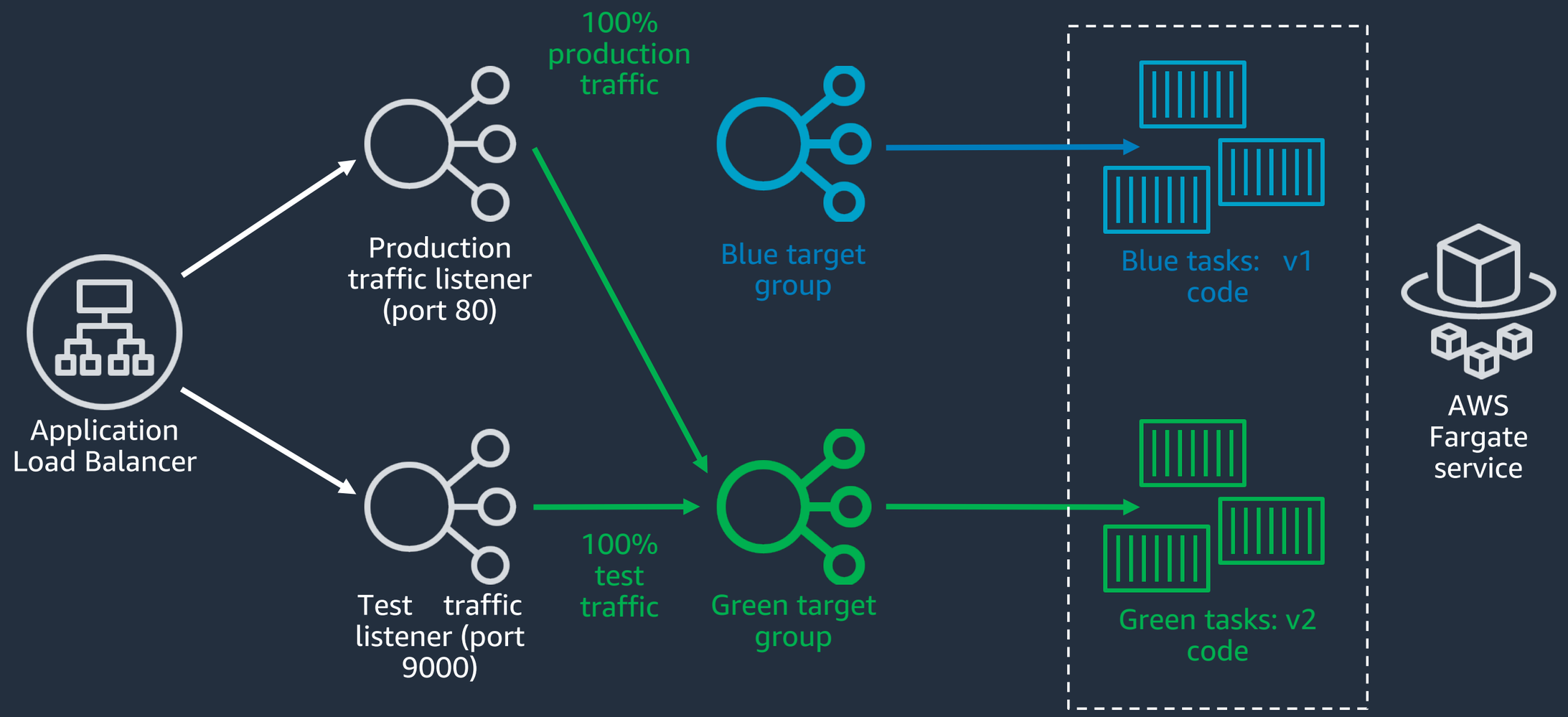
Shift test traffic to green; run validation tests against test endpoint



aws modern apps

CodeDeploy: Amazon ECS blue/green deployment

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



aws modern apps

Container image tagging for deployments

- Docker tags are resolved when each container starts, not just during deployments
- Deploying “latest” or “prod” can result in untested code in production after a scale-out event
- Use unique “immutable” tags for deployments

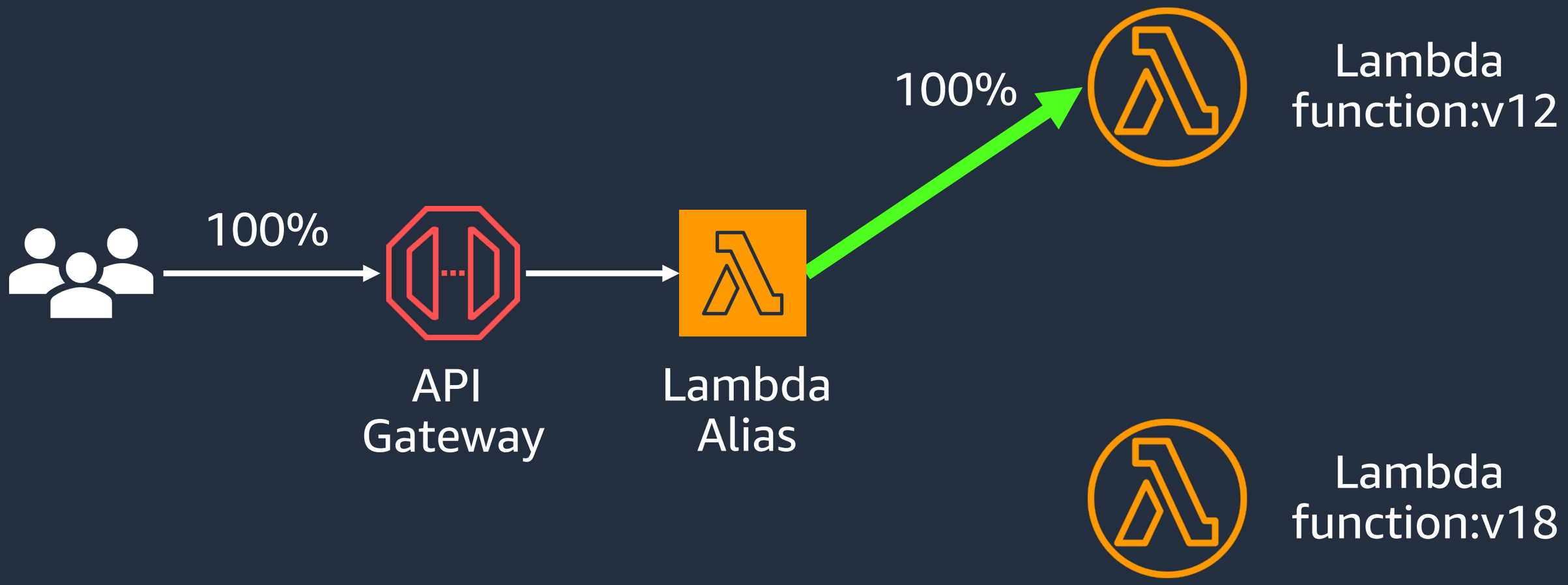
CodeDeploy-Lambda deployments

Enable in your serverless application template

```
Resources:
  GetFunction:
    Type: AWS::Serverless::Function
    Properties:
      DeploymentPreference:
        Type: Canary10Percent10Minutes
        Alarms:
          - !Ref ErrorsAlarm
        Hooks:
          PreTraffic: !Ref PreTrafficHook
```

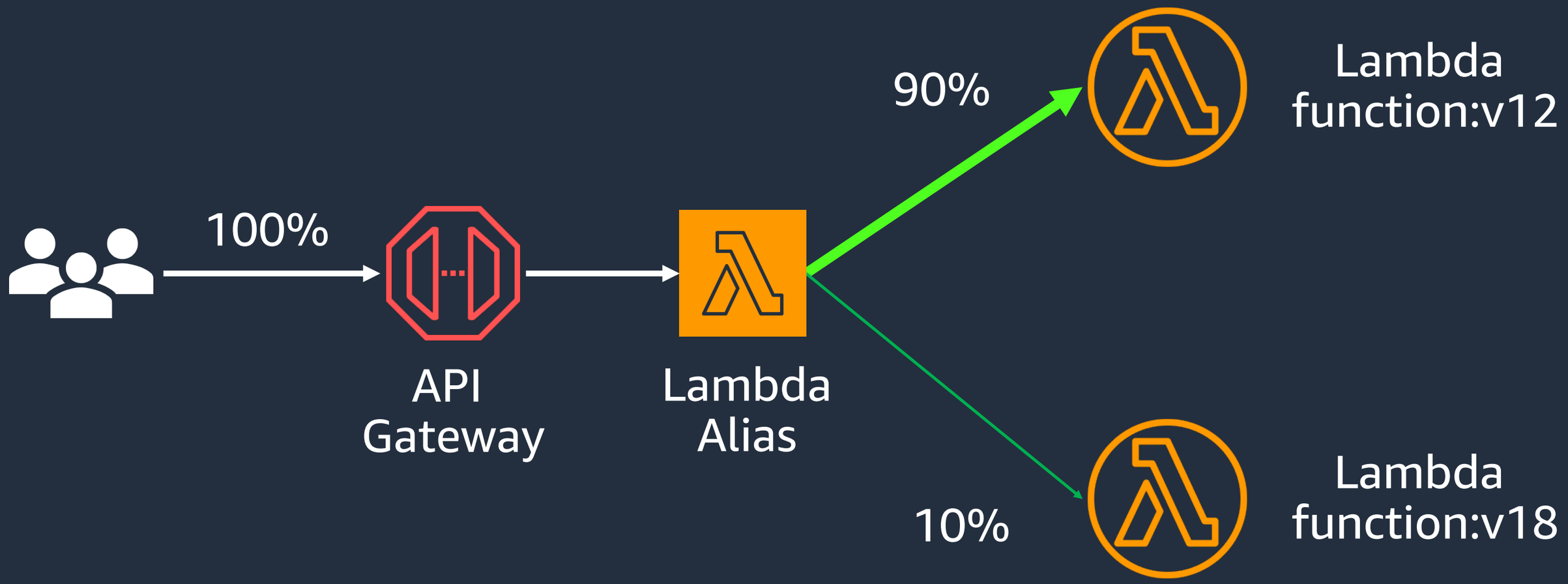

aws modern apps

AWS CodeDeploy: AWS Lambda deployments



aws modern apps

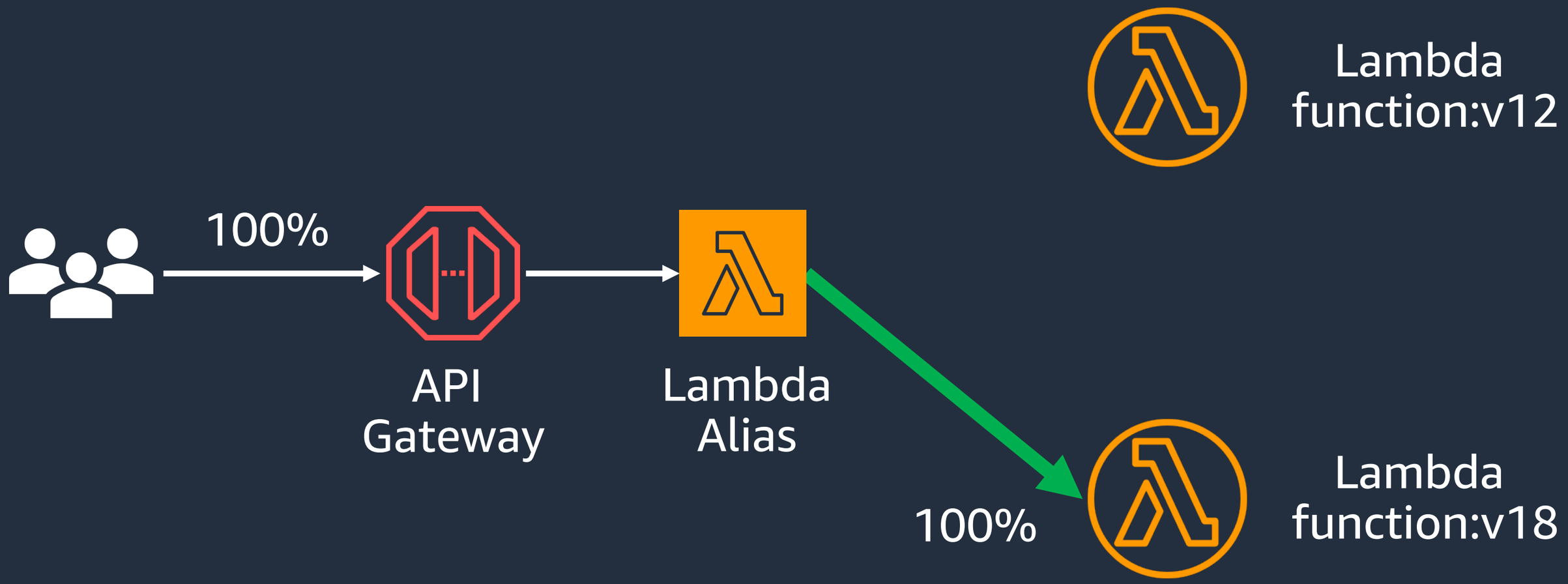
AWS CodeDeploy: AWS Lambda deployments



Canary: “shift 10% of traffic for 10 mins, then shift the rest”

aws modern apps

AWS CodeDeploy: AWS Lambda deployments



Canary: “shift 10% of traffic for 10 mins, then shift the rest”

aws modern apps

Best practices for CI/CD

1

Pipeline Automation

2

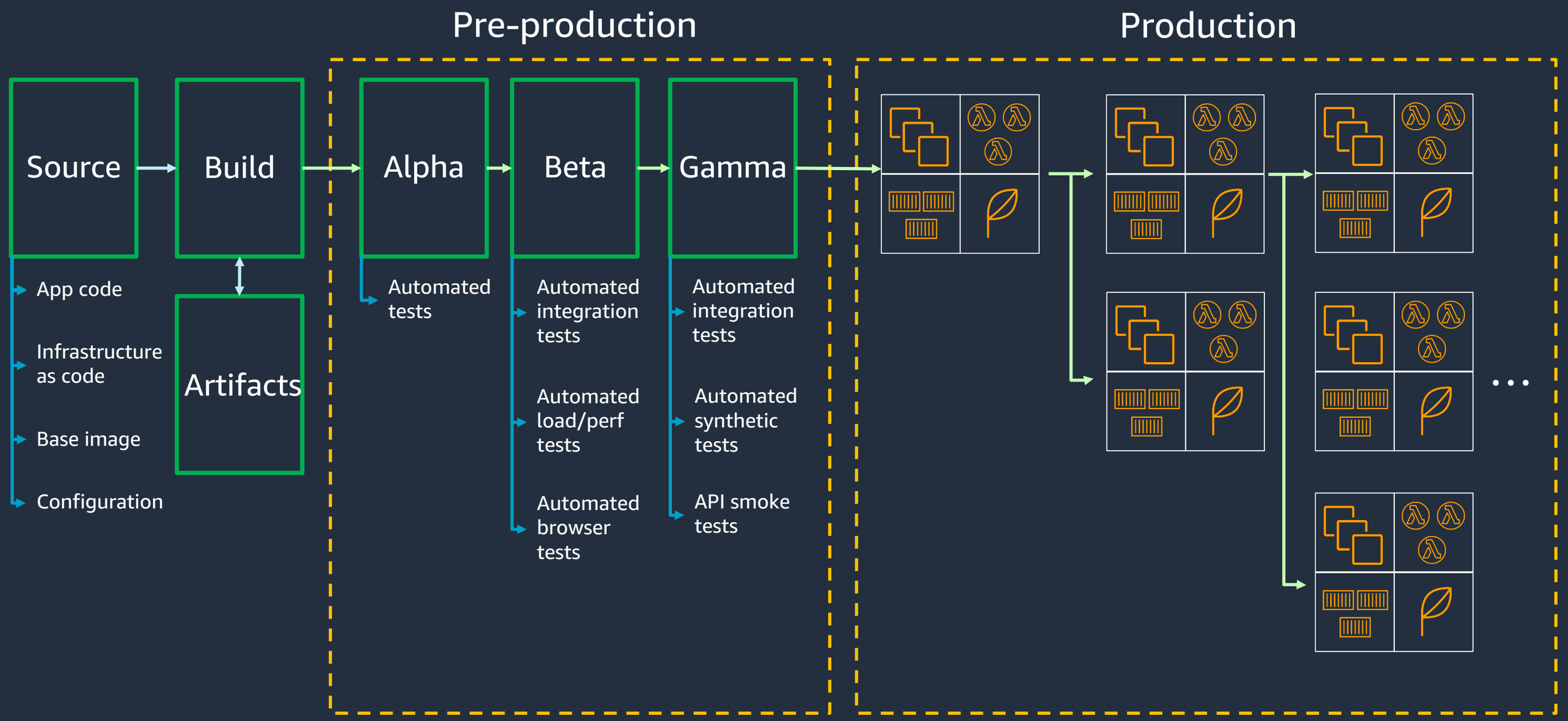
Safe Deployments

3

Repeatable infrastructure changes

aws modern apps

What is DevOps at scale?



aws modern apps

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 modern apps

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 AWS 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 modern apps

CDK pipelines: Construct

```
export class MyMicroservicePipeline extends cdk.Construct {
    constructor(parent: cdk.Construct, name: string, props: MyMicroservicePipelineProps) {
        super(parent, name);

        const pipeline = new codepipeline.Pipeline(this, 'Pipeline', {
            pipelineName: props.serviceName,
        });

        const githubAccessToken = new cdk.SecretParameter(this, 'GitHubToken',
            { ssmParameter: 'GitHubToken' });
        new codepipeline.GitHubSourceAction(this, 'GitHubSource', {
            stage: pipeline.addStage('Source'),
            owner: 'myorg',
            repo: props.serviceName,
            oauthToken: githubAccessToken.value
        });
…
```


aws modern apps

CDK pipelines: Stack

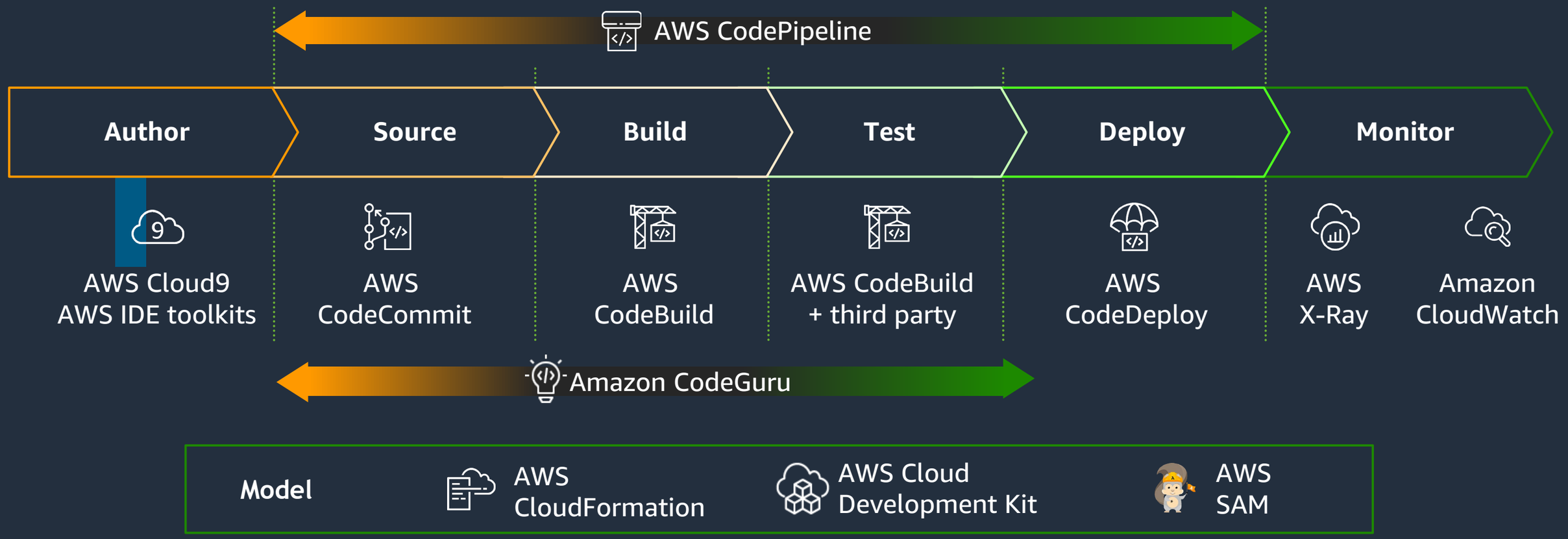
```
import cdk = require('@aws-cdk/cdk');
import { MyMicroservicePipeline } from './pipeline';

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

        new MyMicroservicePipeline(this, 'Pipeline1', { 'serviceName': 'Microservice1' });
        new MyMicroservicePipeline(this, 'Pipeline2', { 'serviceName': 'Microservice2' });
        new MyMicroservicePipeline(this, 'Pipeline3', { 'serviceName': 'Microservice3' });
        new MyMicroservicePipeline(this, 'Pipeline4', { 'serviceName': 'Microservice4' });
    }
}
const app = new cdk.App();
new MyMicroservicePipelinesStack(app, 'MyMicroservicePipelines');
app.run();
```


aws modern apps

CI/CD for modern software delivery



aws modern apps

Visit the Modern Applications Resource Hub for more resources

Dive deeper with these newly created whitepapers and e-books to accelerate your modernization journey.

- Modern Applications e-book
- Accelerating your AWS journey: Migration & Modernization
- Journey to serverless-first report
- Modernize today with containers on AWS
- … and more!

https://tinyurl.com/aws-modern-apps

Visit resource hub »

aws modern apps

Accelerate Your Modernization Journey

Develop skills in designing, building, and managing modern applications

90% of IT decision makers report cloud skills shortages[1]. A lack of cloud skills impacts modern application development. Start your modern application development journey with AWS Training & Certification.

Take free digital training

With a little time and initiative, learners can enhance their practical cloud knowledge through free digital training. These on-demand courses, which vary in length from 10 minutes to several hours, can help one broaden their understanding of specific subjects such as serverless, containers, and developer tools.

Get live, hands-on, instructor-led training

Whether physical or virtual, classroom training offers more in-depth instruction for people who want to deepen their technical skills. Classes are a mix of presentations, hands-on labs, and group discussions led by experts in their fields. Courses include Developing on AWS and Advanced Developing on AWS.

Quickly ramp up your modern application skills

Independent learning allows people to fill in knowledge gaps and learn new topics at their own pace. There's a wide range of whitepapers, blog posts, videos, webinars, use cases, and peer resources available for IT professionals who want to dive deep into specific technical topics. Learn more.

[1] 451 Research, *Demystifying Cloud Transformation: Where Enterprises Should Start*, September 2019.

aws modern apps

Thank you for attending AWS Modern Applications 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-apac-marketing@amazon.com

twitter.com/AWSCloud

facebook.com/AmazonWebServices

youtube.com/user/AmazonWebServices

slideshare.net/AmazonWebServices

twitch.tv/aws

aws modern apps

aws modern apps

Thank you!