AWS re:Invent

SVS328-R

Build a CI/CD pipeline for serverless applications

Nitin Vashishtha
Solutions Architect
Amazon Web Services

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

What are we going to build?

- Develop a Lambda function with SAM
- Develop a CI/CD pipeline
- Automate Lambda function deployments with traffic shifting

Approaches to modern application development



AWS Lambda

Serverless functions

Event-driven
Many language runtimes
Data source integrations
No server management



AWS Fargate

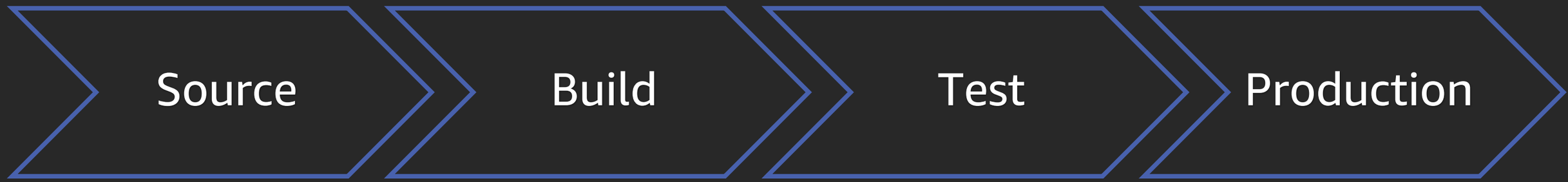
Serverless containers

Long-running
Abstracts the OS
Fully managed orchestration
Fully managed cluster scaling

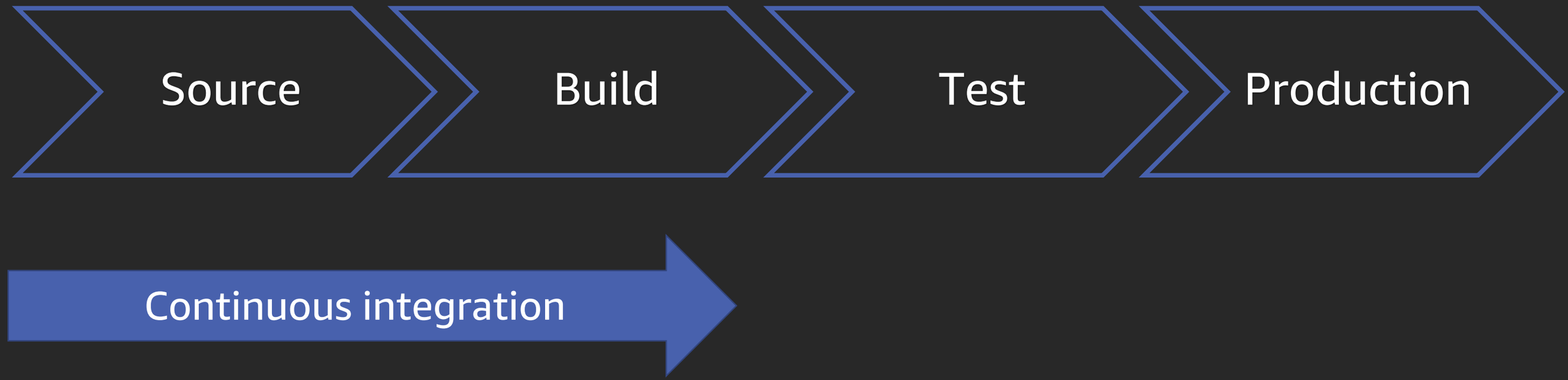
Approaches to modern application development

- Simplify environment management with serverless technologies
- Reduce the impact of code changes with microservice architectures
- Automate operations by modeling applications & infrastructure as code
- Accelerate the delivery of new, high-quality services with CI/CD
- Gain insight across resources and applications by enabling observability
- Protect customers and the business with end-to-end security & compliance

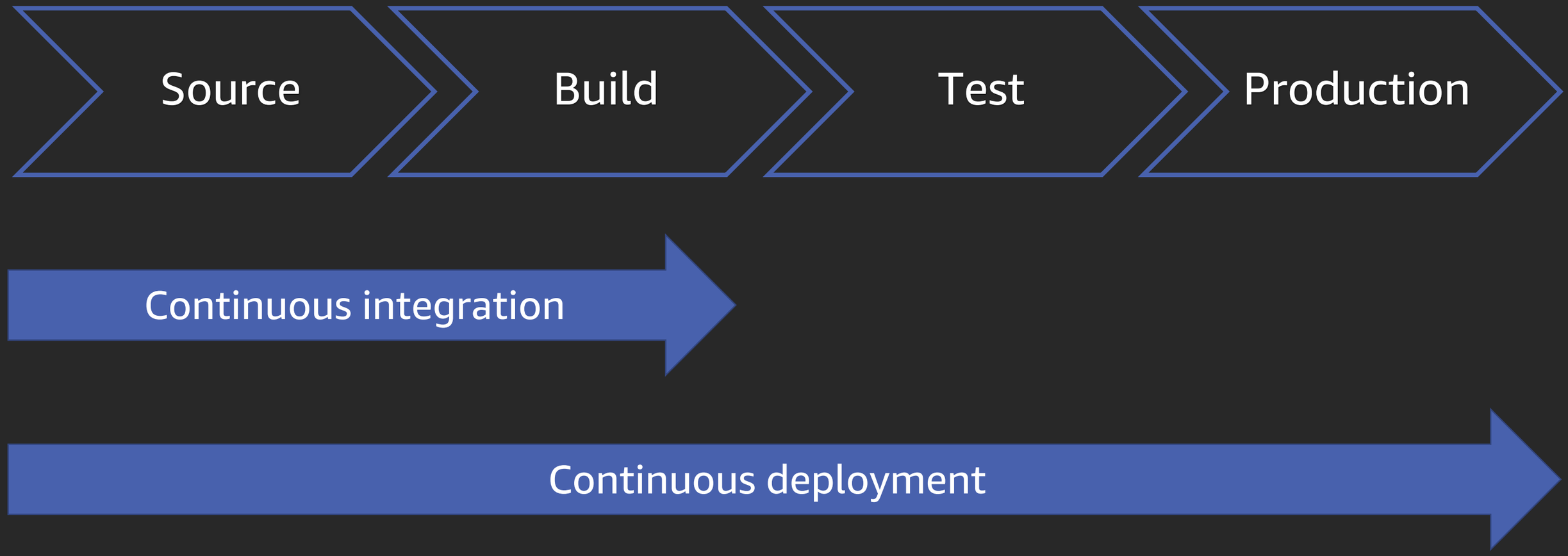
Release process stages



Release process stages



Release process stages

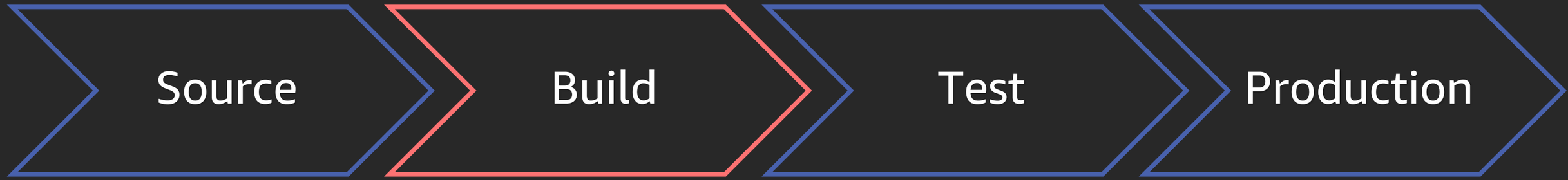


Release process stages



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

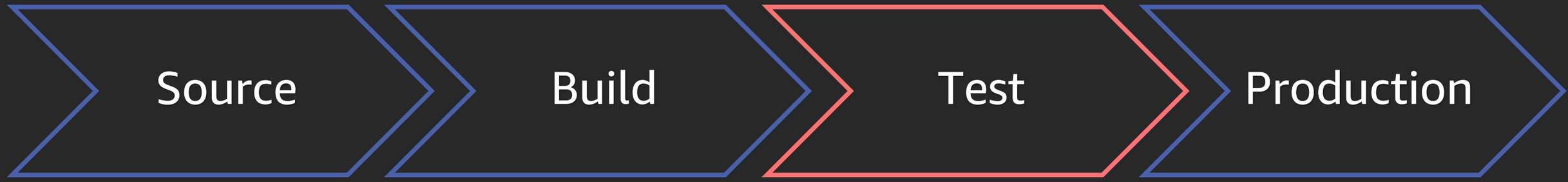
Release process stages



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

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

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 function deployment packages

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

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 function deployment packages

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

- Deployment to production environments
- Monitor code in production to quickly detect errors

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 function deployment packages

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

- Deployment to production environments
- Monitor code in production to quickly detect errors

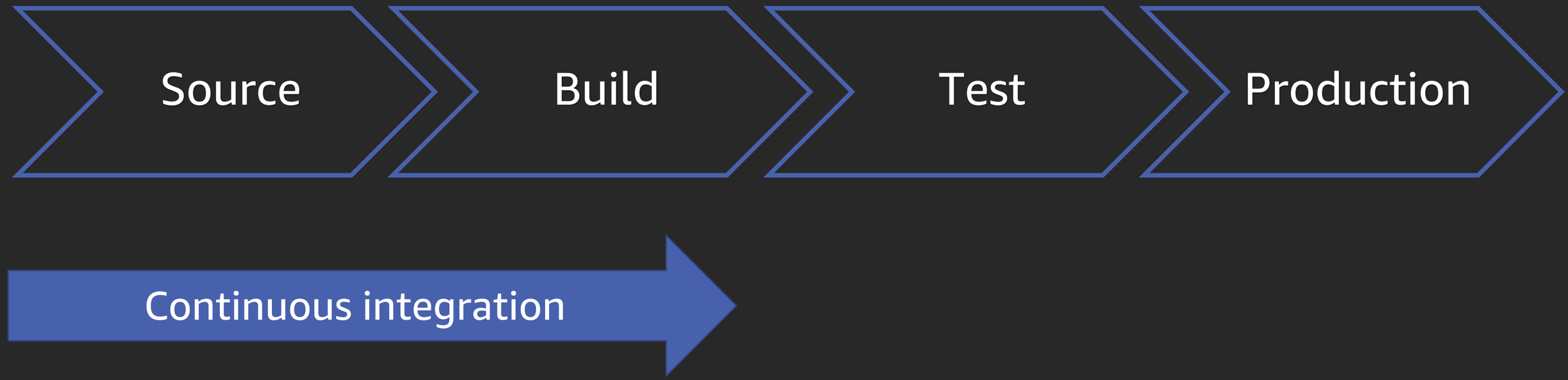
Pillars of releasing modern applications

Continuous integration

Continuous deployment

Infrastructure as code

Continuous integration goals



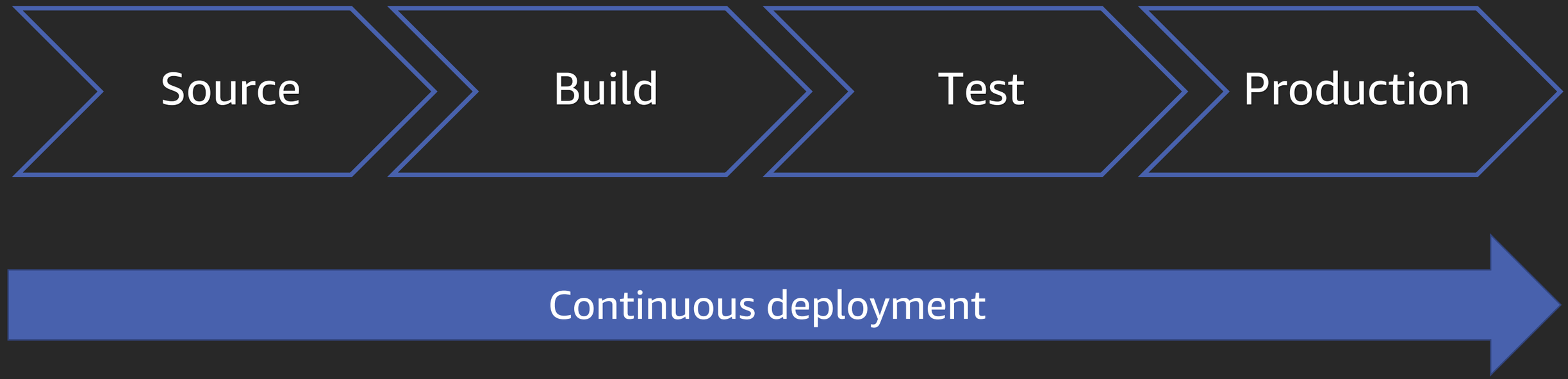
Pillars of releasing modern applications

Continuous integration

Continuous deployment

Infrastructure as code

Continuous deployment goals



AWS CodeDeploy



- Automates code deployments to any instance and Lambda
- Handles the complexity of updating your applications
- Avoid downtime during application deployment
- Roll back automatically if failure detected
- Deploy to Amazon EC2, Lambda, or on-premises servers

CodeDeploy-Lambda deployments

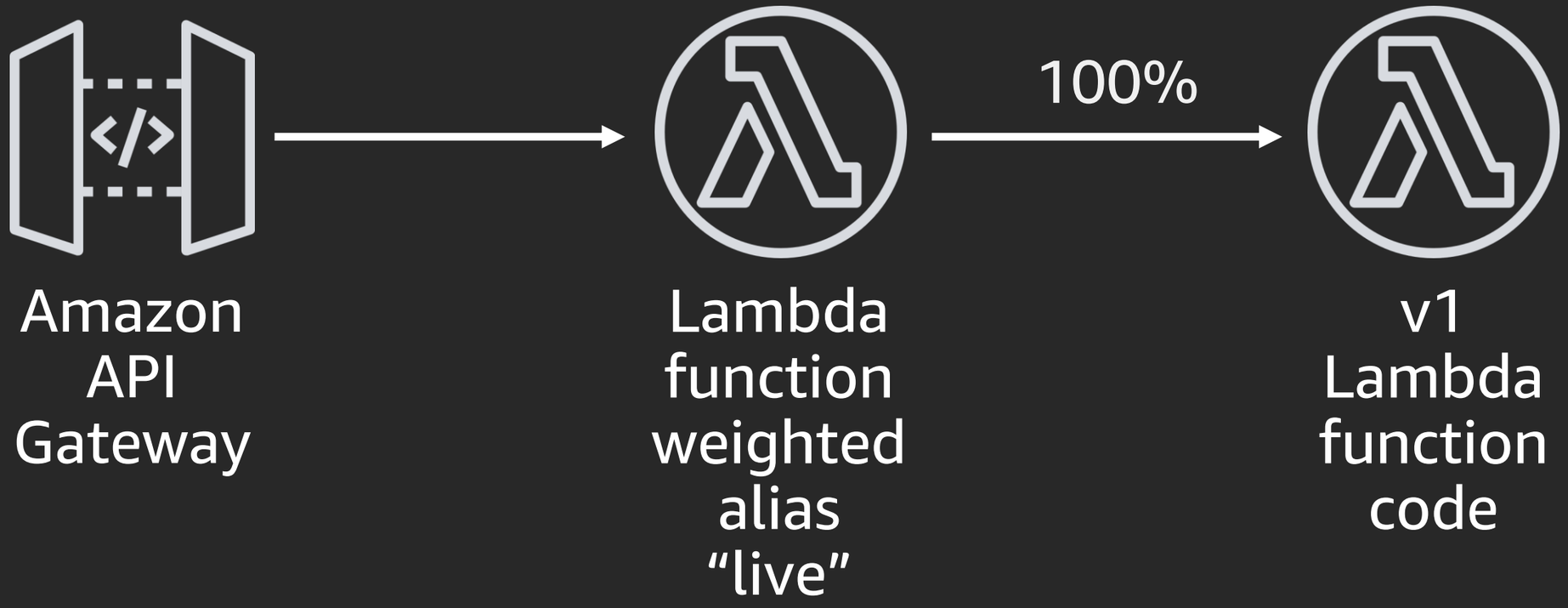
- Shifts traffic using Lambda function weighted aliases
- Choose canary (“shift 10% of traffic for 10 minutes, then shift rest”) or linear (“shift 10% more traffic every 10 minutes”)
- Validation “hooks” enable testing at each stage of the deployment
- Fast rollback in seconds if case of hook failure or Amazon CloudWatch alarms
- Monitor deployment status and history via console, API, Amazon Simple Notification Service (Amazon SNS) notifications, and Amazon CloudWatch Events

CodeDeploy-Lambda deployments

Enable in your serverless application template

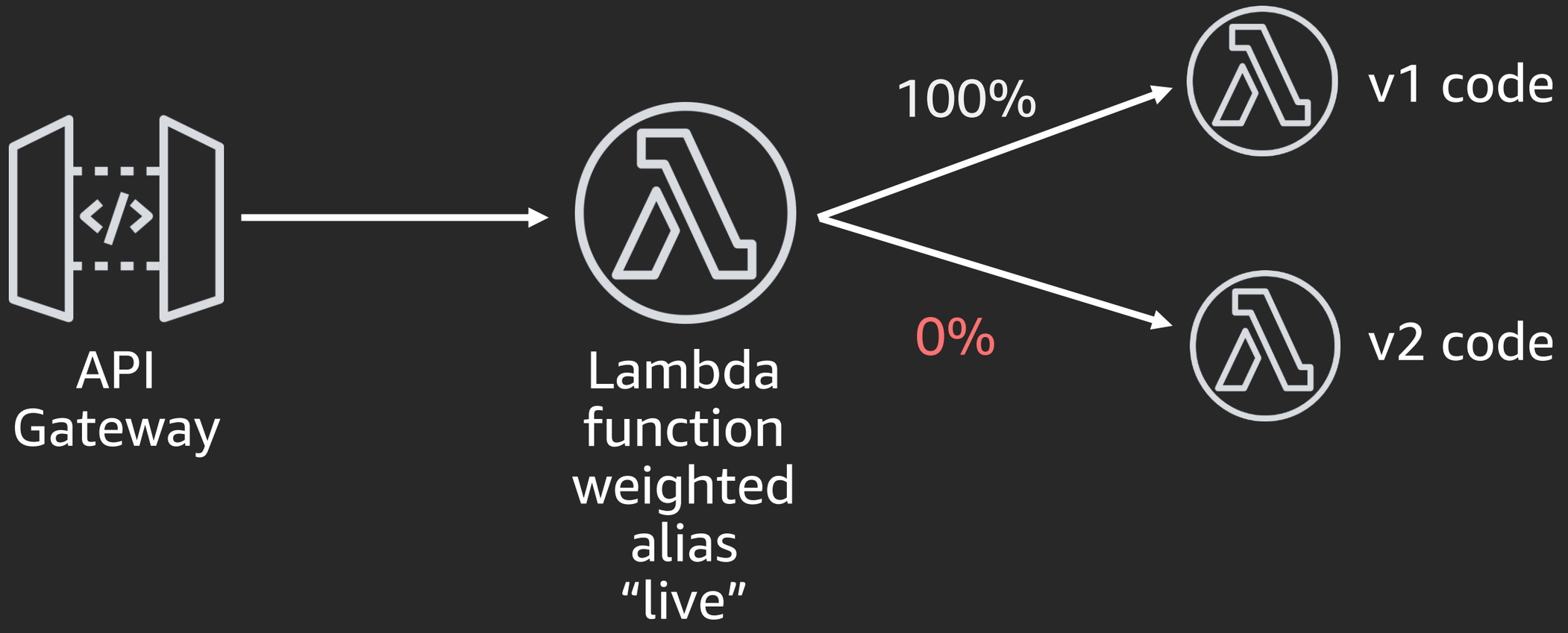
```
Resources:
  GetFunction:
    Type: AWS::Serverless::Function
    Properties:
      DeploymentPreference:
        Type: Canary10Percent10Minutes
        Alarms:
          - !Ref ErrorsAlarm
        Hooks:
          PreTraffic: !Ref PreTrafficHook
```

CodeDeploy-Lambda canary deployment



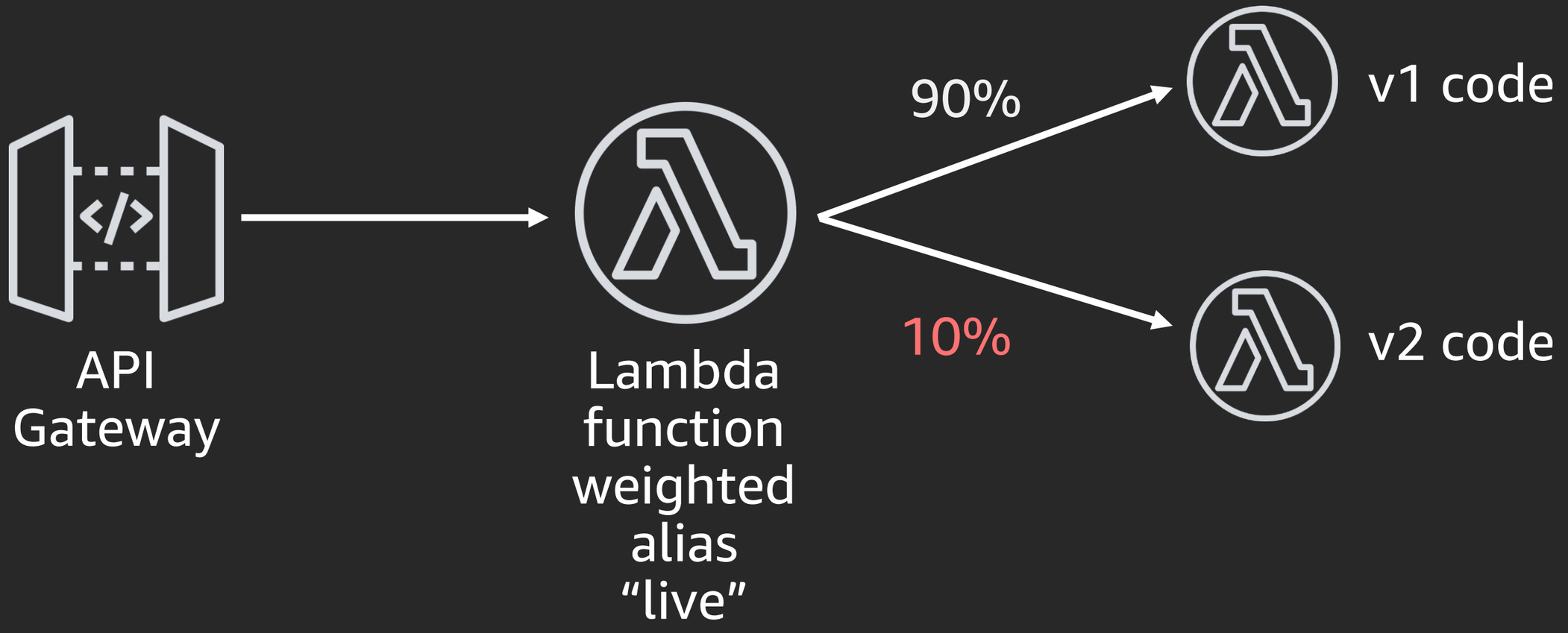
CodeDeploy-Lambda canary deployment

Run hook against v2 code before it receives traffic



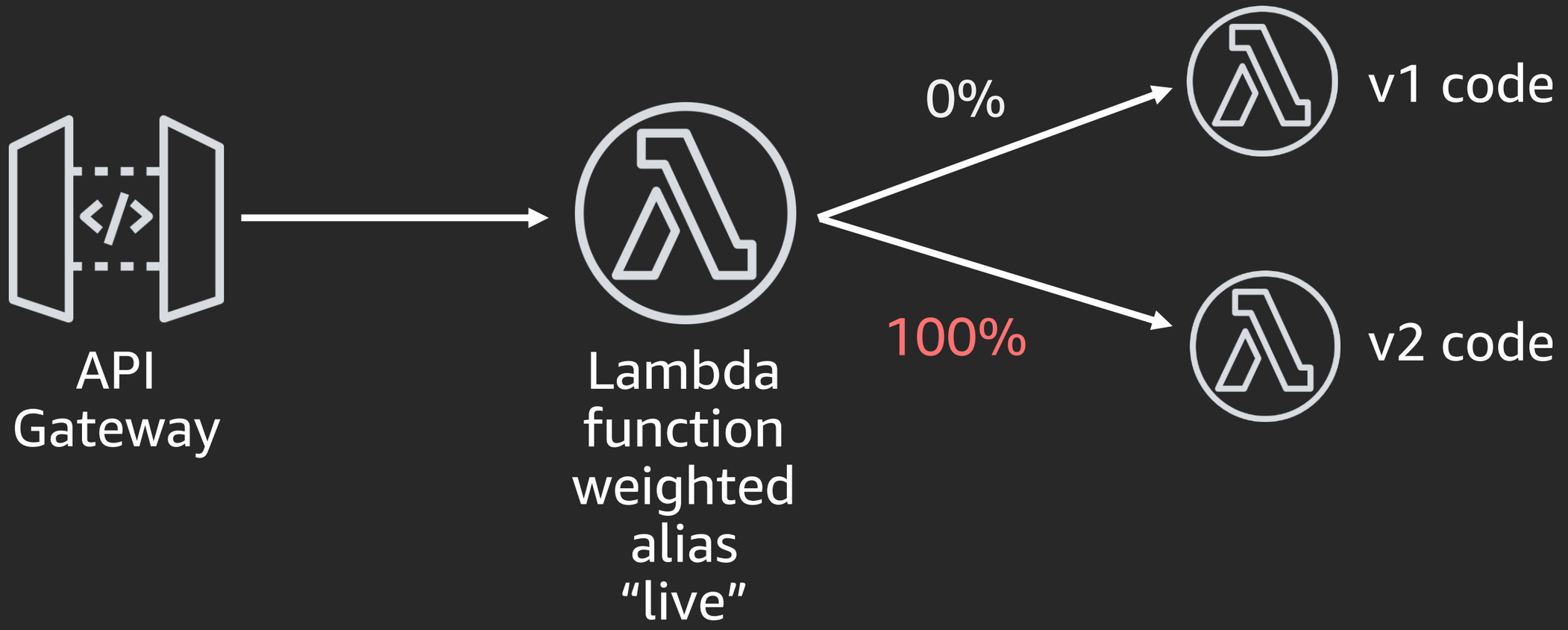
CodeDeploy-Lambda canary deployment

Wait for 10 minutes, roll back in case of alarm



CodeDeploy-Lambda canary deployment

Complete deployment



Model function environments with AWS Serverless Application Model (SAM)

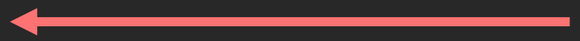


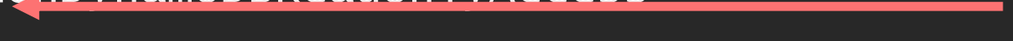
- Open-source framework for building serverless applications on AWS
- Shorthand syntax to express functions, APIs, databases, and event source mappings
- Transforms and expands SAM syntax into AWS CloudFormation syntax on deployment
- Supports all AWS CloudFormation resource types

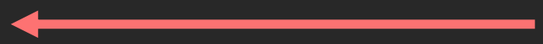
https://aws.amazon.com/serverless/sam/

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  GetFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.get
      Runtime: nodejs6.10
      CodeUri: src/
      Policies: AmazonDynamoDBReadOnlyAccess
      Events:
        GetResource:
          Type: Api
          Properties:
            Path: /resource/{resourceId}
            Method: get
```







Shorthand syntax to express functions, tables, and events

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  GetFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.get
      Runtime: nodejs6.10
      CodeUri: src/
      Policies: AmazonDynamoDBReadOnlyAccess
      Events:
        GetResource:
          Type: Api
          Properties:
            Path: /resource/{resourceId}
            Method: get
```

Just 18 lines of SAM syntax creates many resources:

- Lambda function
- API Gateway
- Amazon DynamoDB table
- AWS Identity and Access Management (IAM) roles

Use SAM CLI to package and deploy SAM templates

```
pip install --user aws-sam-cli
```

```
sam init
```




```
sam build
```

```
sam package
```

```
sam deploy
```

AWS CodePipeline

Use AWS CloudFormation deployment actions with any SAM application

Jenkins

Use SAM CLI plugin

Model container environments with AWS Cloud Development Kit (CDK)

AWS CDK

- Open-source framework to define cloud infrastructure in Typescript
- Provides library of higher-level resource types (“construct” classes) that have AWS best practices built in by default, packaged as npm modules
- Provisions resources with AWS CloudFormation
- Supports all AWS CloudFormation resource types

https://github.com/aws/aws-cdk

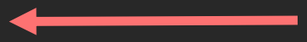
CDK template

```
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"),
    });
  }
}
```



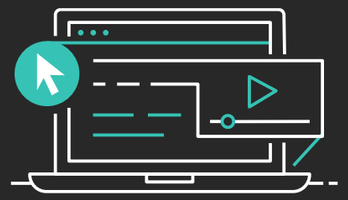
High-level VPC class includes VPC, subnets, security groups, internet gateway, NAT gateways, and route tables

Let’s build

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

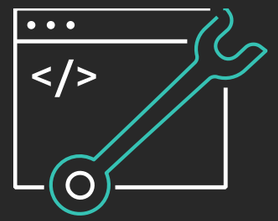
Learn serverless with AWS Training and Certification

Resources created by the experts at AWS to help you learn modern application development



Free, on-demand courses on serverless, including

- Introduction to Serverless Development
- Getting into the Serverless Mindset
- AWS Lambda Foundations
- Amazon API Gateway for Serverless Applications
- Amazon DynamoDB for Serverless Architectures



Additional digital and classroom trainings cover modern application development and computing

Visit the Learning Library at https://aws.training

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws training and certification

Thank you!

Nitin Vashishtha

nvashish@amazon.com

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.