AWS re:Invent

DOP209-R

Introduction to DevOps on AWS

Sébastien Stormacq
Senior Developer Advocate
Amazon Web Services, EMEA

Jonathan Weiss
Senior Manager, AWS Development Tools
Amazon Web Services

AWS re:Invent

aws

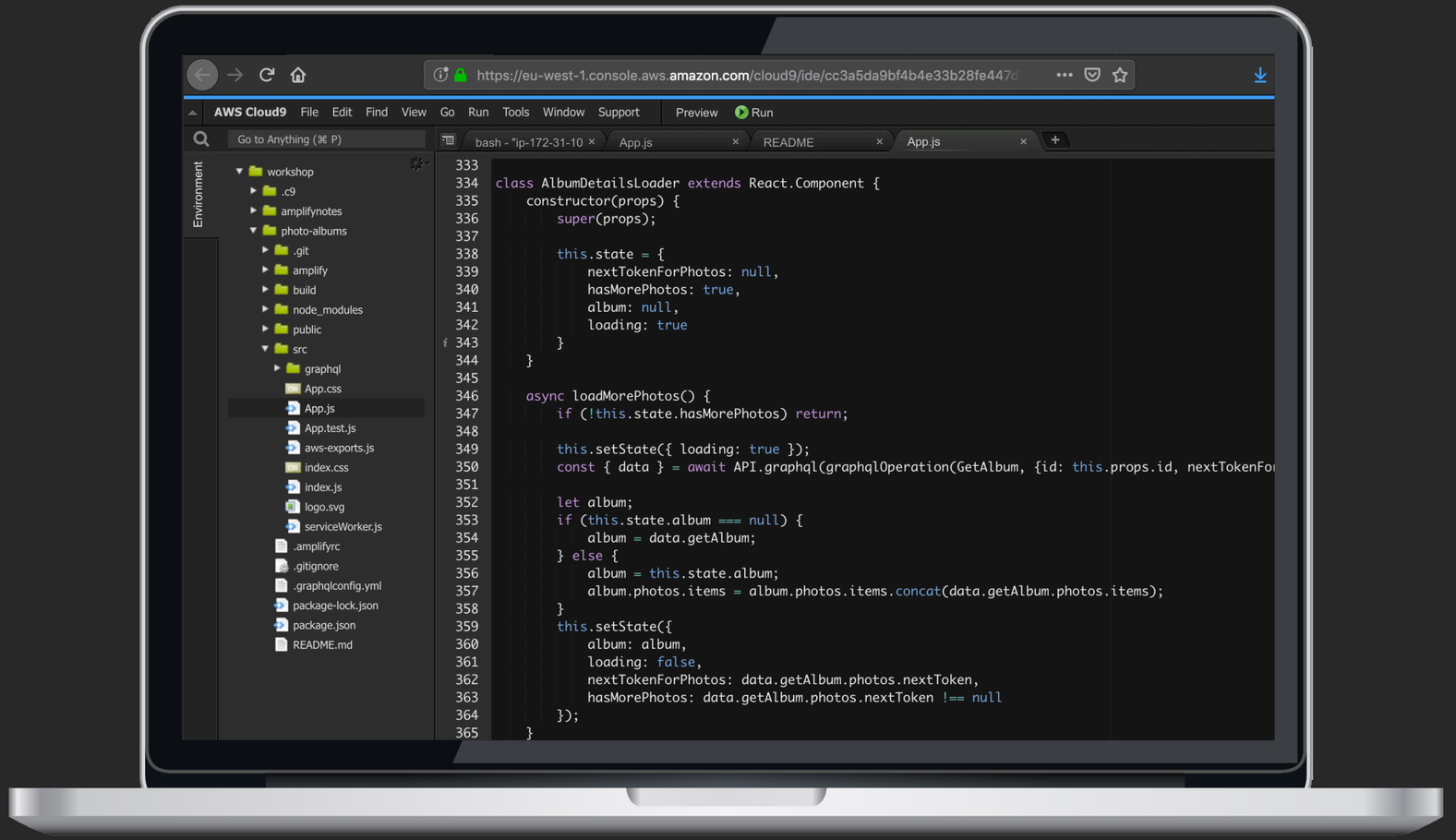
Day 1

AWS re:Invent

aws

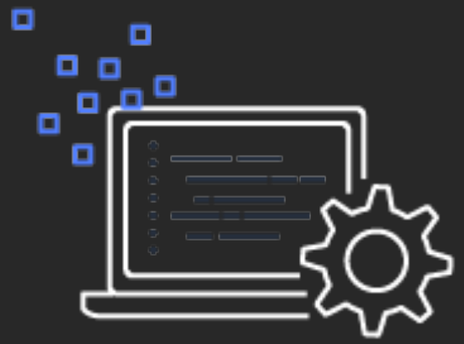
It’s always day 1

How it all started?

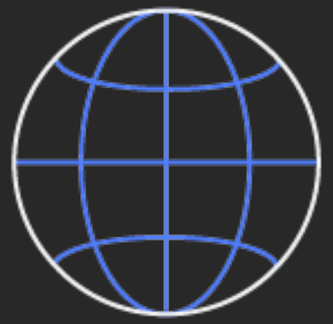


Phase 1

Cloud benefits



Agility

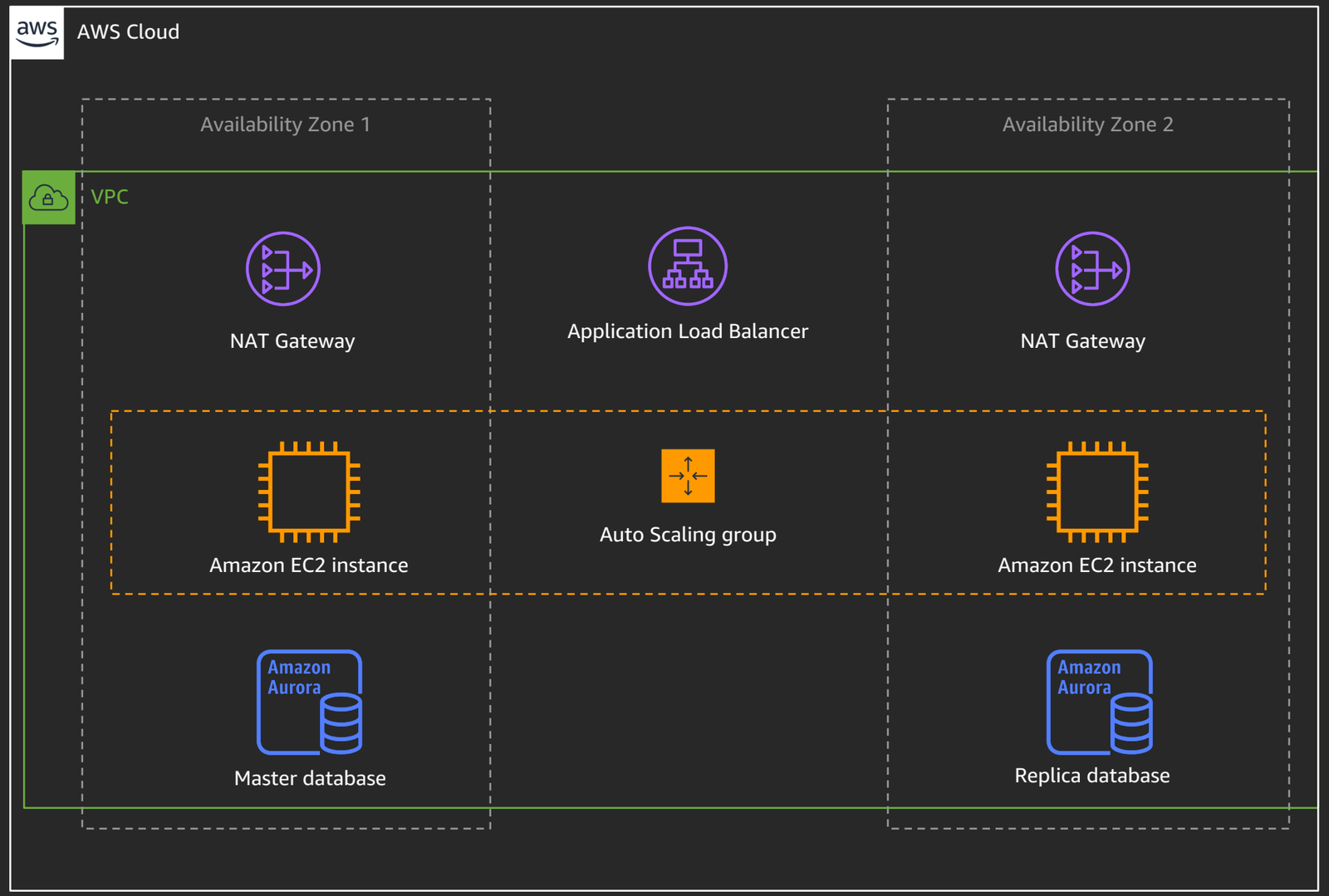


Go global

Elasticity

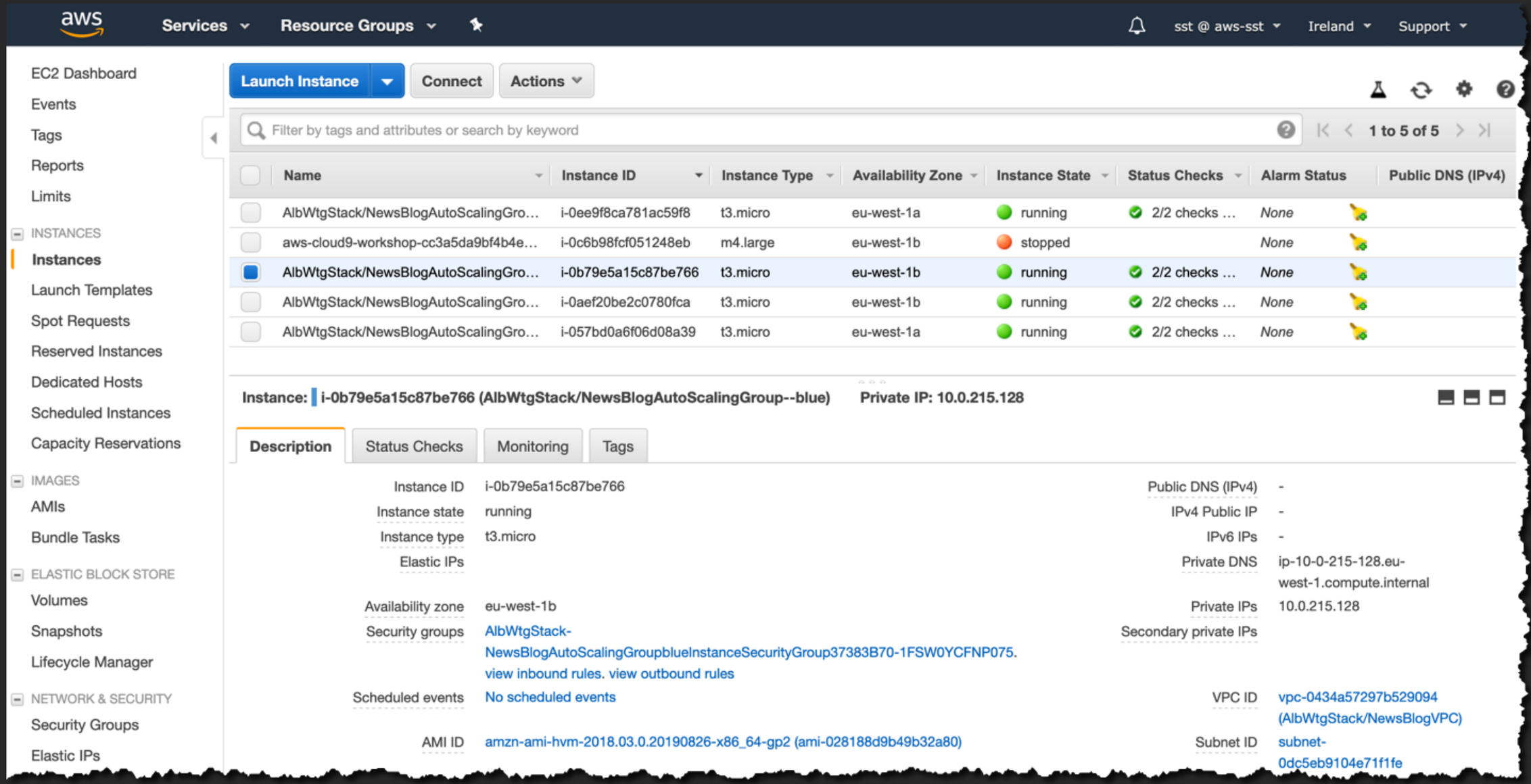
Pay for the value

References architectures



https://aws.amazon.com/architecture/

Infrastructure as click



You choose your IDE

AWS Toolkit
for PyCharm

Python

AWS Toolkit
for IntelliJ

Java, Python

AWS Toolkit for
Visual Studio Code

.NET, Node

AWS Toolkit for
Visual Studio

.NET

You choose your IDE

AWS Toolkit
for PyCharm

Python

AWS Toolkit
for IntelliJ

Java, Python

AWS Toolkit for
Visual Studio Code

.NET, Node

AWS Toolkit for
Visual Studio

.NET

New

AWS Toolkit
for Webstorm

Node.js

AWS Toolkit
for Rider

.NET

```
69
70          # api-gateway-simple-proxy-for-lambda-output-format
71          https: // docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-inte
72      """
73
74      try:
75          ip = requests.get("http://checkip.amazonaws.com/")
76      except requests.RequestException as e:
77          # Send some context about this error to Lambda Logs
78          print(e)
79
80          raise e
81
82      return {
83          "statusCode": 200,
84          "body": json.dumps(
85              {"message": "hello world", "location": ip.text.replace("\n", "")}
86          ),
87      }
88
```

Your IDE is in the cloud

AWS Cloud9

A cloud IDE for writing, running, and debugging code

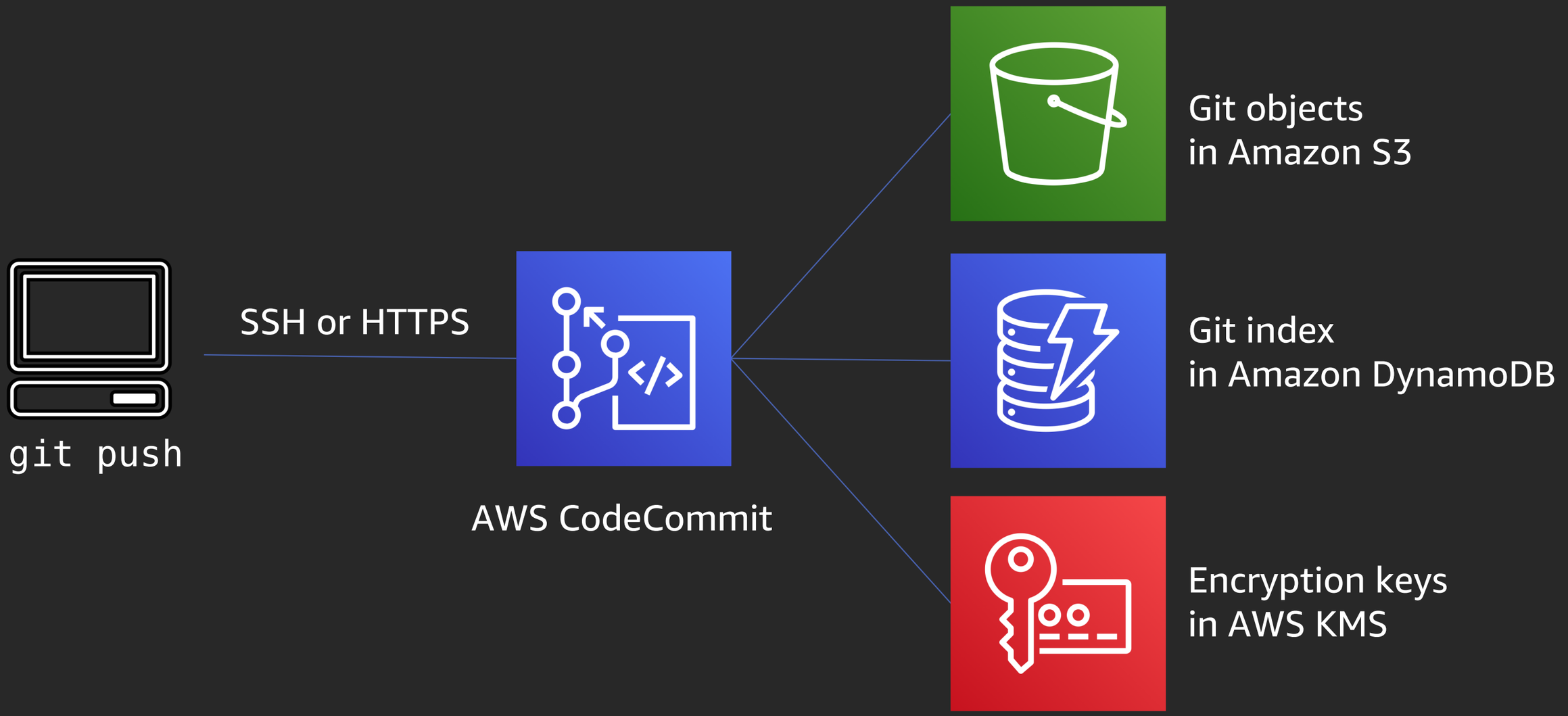
- Code with just a browser
- Start new projects quickly
- Code together in real time
- Build serverless applications with ease
- Direct terminal access

Phase 2

You move your code to a code repository

Secure, scalable, and managed Git source control

You move your code to a code repository



Secure, scalable, and managed Git source control

Getting started with CodeCommit & ssh

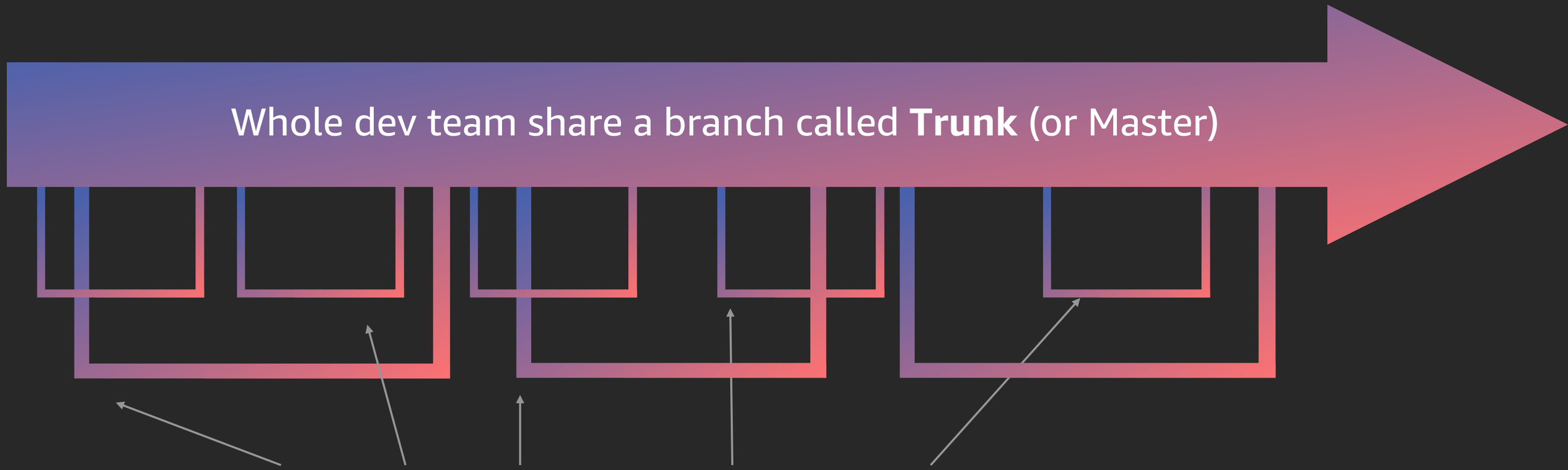
```
$ ssh-keygen
```


```
$ vi ~/.ssh
```

```
    Host git-codecommit.*.amazonaws.com
         User APKAEiBAERJR2EXAMPLE
         identityFile ~/.ssh/codecommit_rsa
```

```
$ git clone                                    \
    ssh://git-codecommit.<region>.amazonaws.com/v1/repos/<repo> \
    <dir>
```

Branching strategy

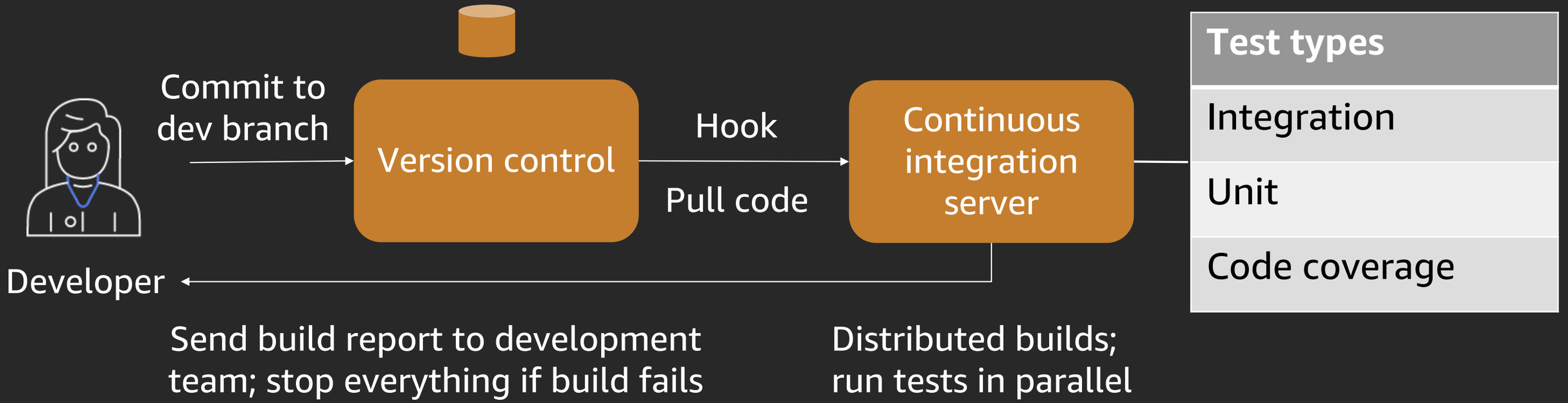


Developers create short-lived feature branches,

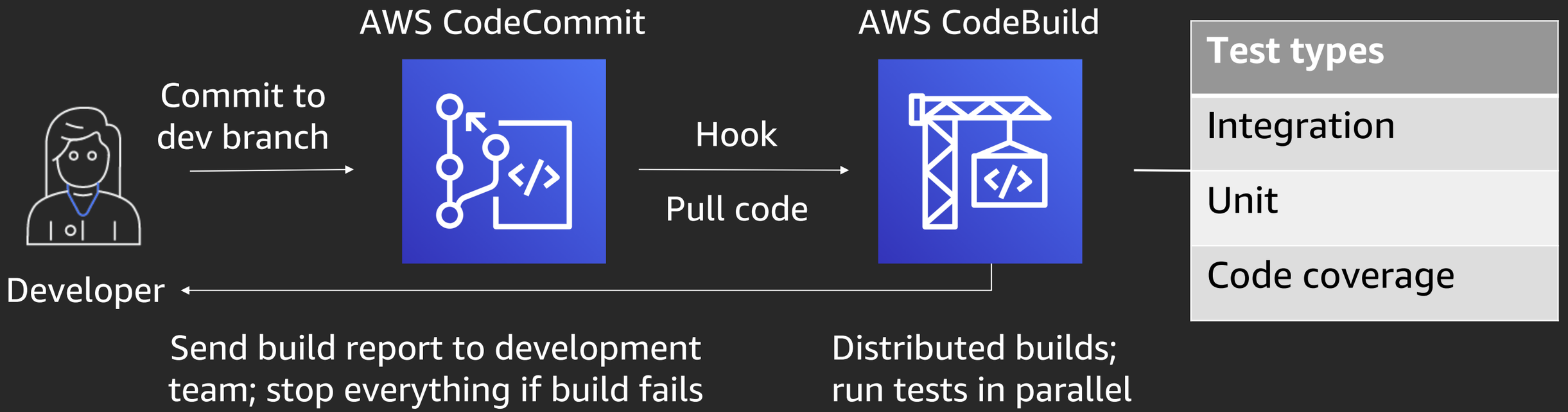
merged through pull requests

Phase 3

Continuous integration workflow



Continuous integration workflow



Anatomy of a buildspec file

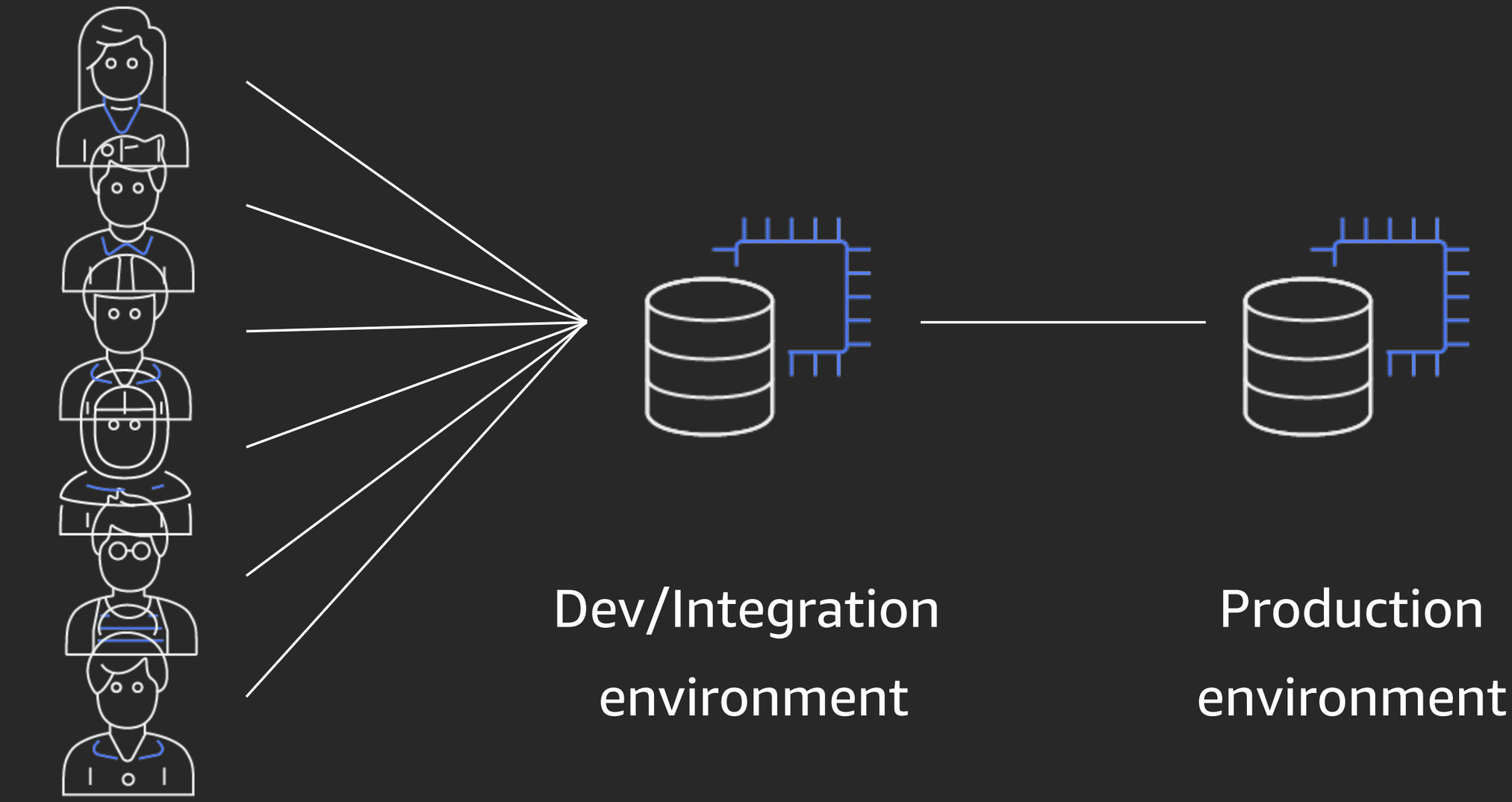
```
version: 0.2

phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws --version
      - $(aws ecr get-login --region eu-west-1 --no-include-email)
      - REPOSiTORY_URi=486652066693.dkr.ecr.eu-west-1.amazonaws.com/nginx
      - iMAGE_TAG=$(echo $CODEBUiLD_RESOLVED_SOURCE_VERSiON | cut -c 1-7)
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $REPOSiTORY_URi:latest nginx/.
      - docker tag $REPOSiTORY_URi:latest $REPOSiTORY_URi:$iMAGE_TAG
```

Anatomy of a buildspec file

```
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker images...
      - docker push $REPOSiTORY_URi:latest
      - docker push $REPOSiTORY_URi:$iMAGE_TAG
      - echo Writing image definitions file...
      - printf '[{"name":"nginx","imageUri":"%s"}]'
               $REPOSiTORY_URi:$iMAGE_TAG > imagedefinitions.json
artifacts:
    files: imagedefinitions.json
```

Phase 4

One dev environment does not scale

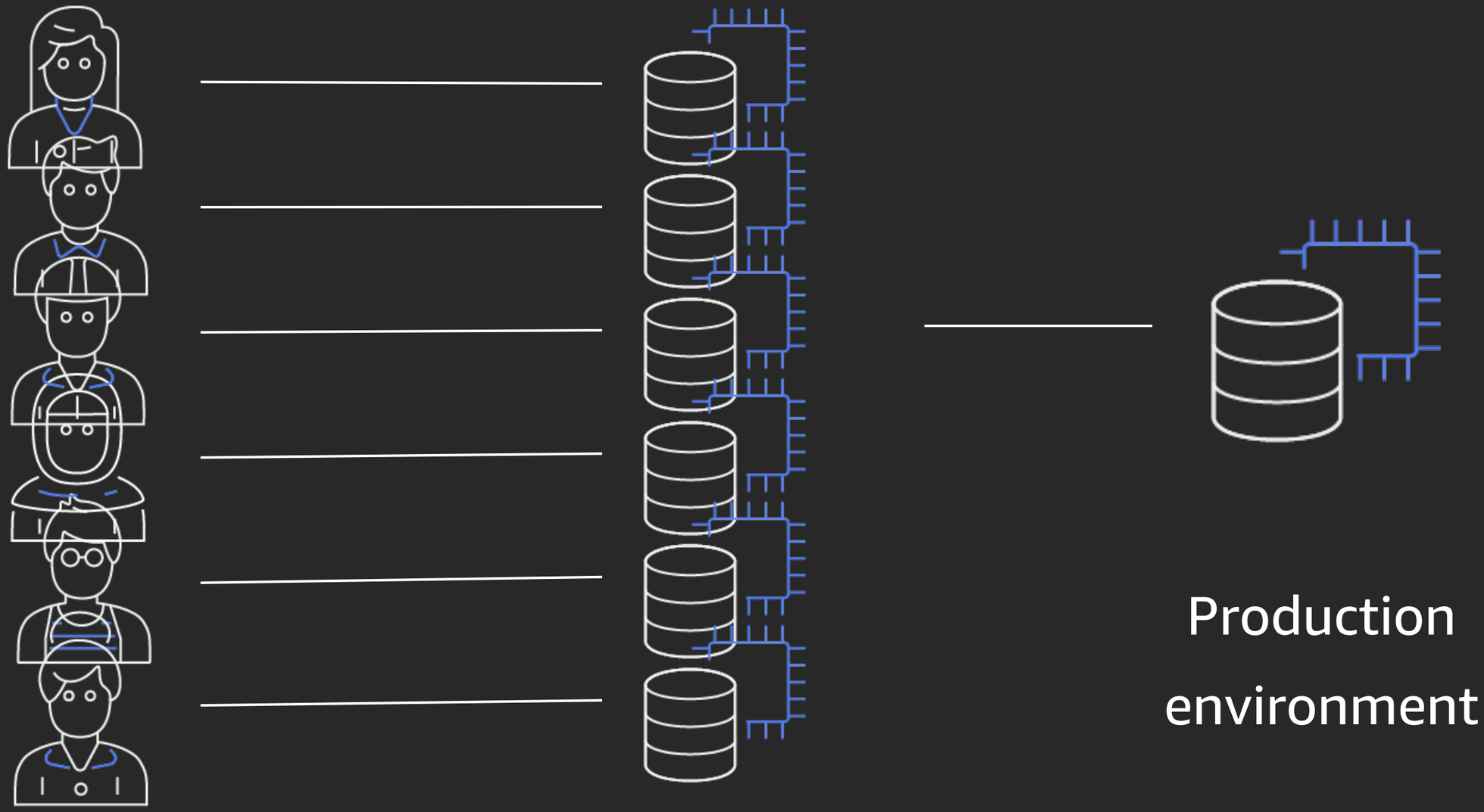


Dev team

Dev/Integration environment

Production environment

One dev/integration environment per developer



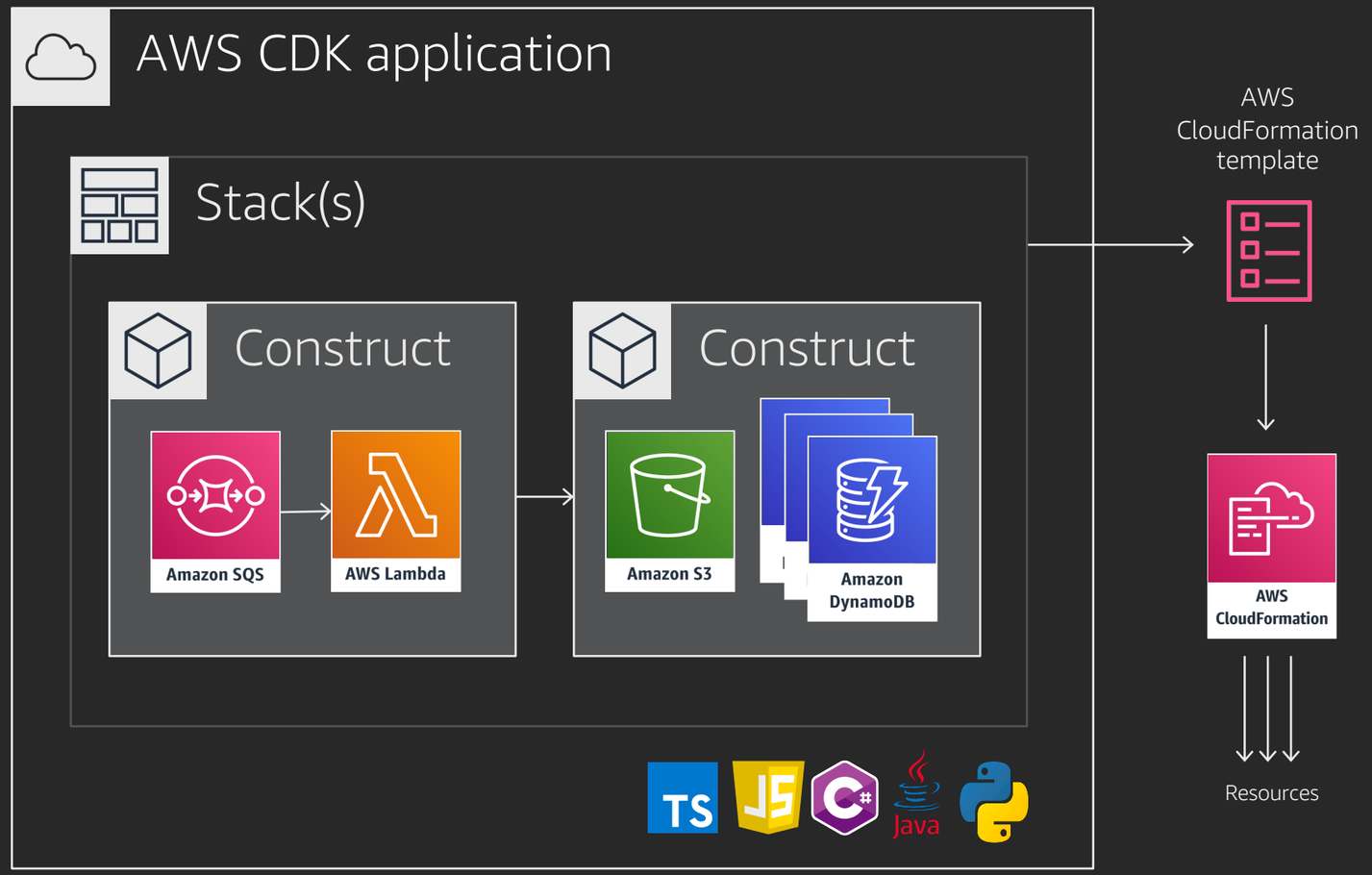
Dev team

Dev/Integration environments

Production environment

Infrastructure as code—avoid snowflakes

Cloud Development Kit (CDK)



CDK: Create a VPC

```
//
// create VPC w/ public and private subnets in 2 AZ
// this also creates a NAT Gateway
//
const vpc = new ec2.Vpc(this, 'NewsBlogVPC', {
  maxAzs : 2
});
```

CDK: Package your application

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

CDK: Bootstrap your servers

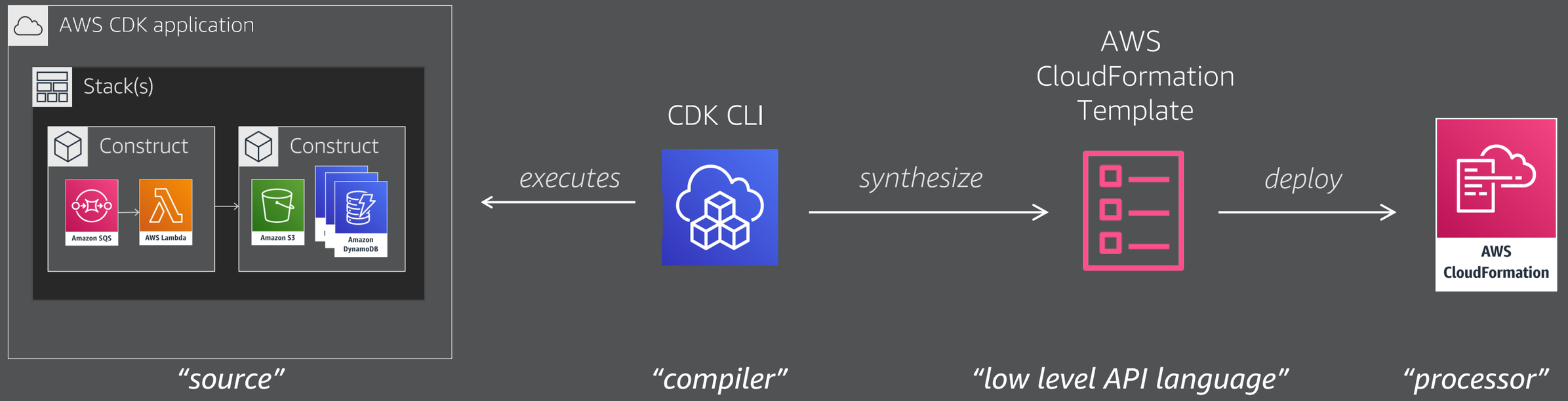
```
// define a user data script to install & launch our app
const userData = UserData.forLinux();

userData.addCommands('yum install -y nginx',
                     'chkconfig nginx on', 'service nginx start');

userData.addCommands(`aws s3 cp s3://${asset.s3BucketName}/${asset.s3ObjectKey} .`,

                     `unzip *.zip`,
                     `/bin/cp -r -n ${env}/* /usr/share/nginx/html/`);
```

CDK: Create an Auto Scaling group

```
  // create an auto scaling group for each environment
  const asg = new autoscaling.AutoScalingGroup(this, 'YourAppgAutoScalingGrou
' , {

    vpc,
    instanceType: ec2.instanceType.of(ec2.instanceClass.BURSTABLE3,
                                      ec2.instanceSize.MiCRO),
    machineimage: new ec2.AmazonLinuximage(),
    desiredCapacity: 2,
    role: role,
    userData: userData
  });
```

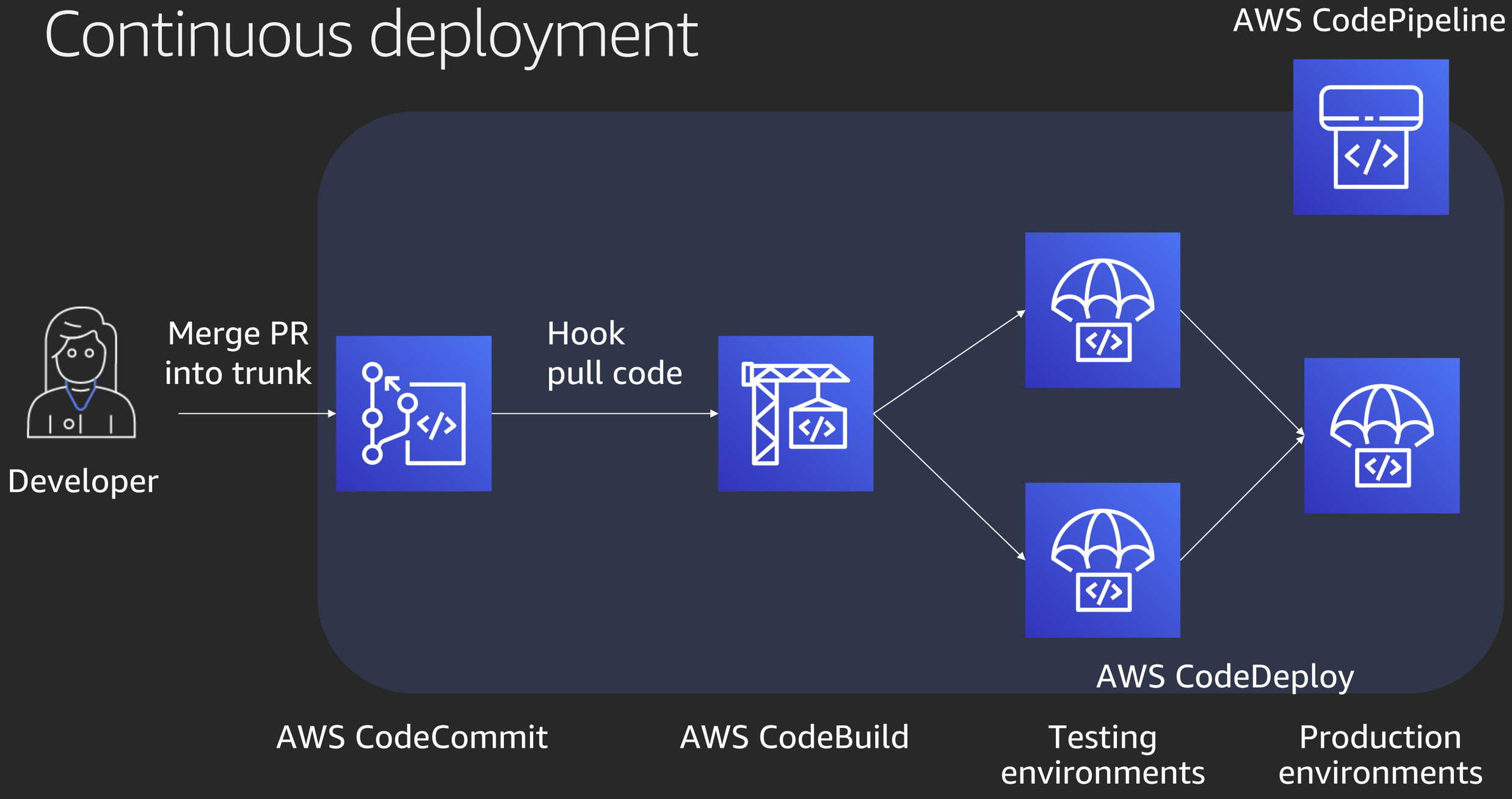
CDK: Deploy your own dev environment

```
$ cdk deploy
```

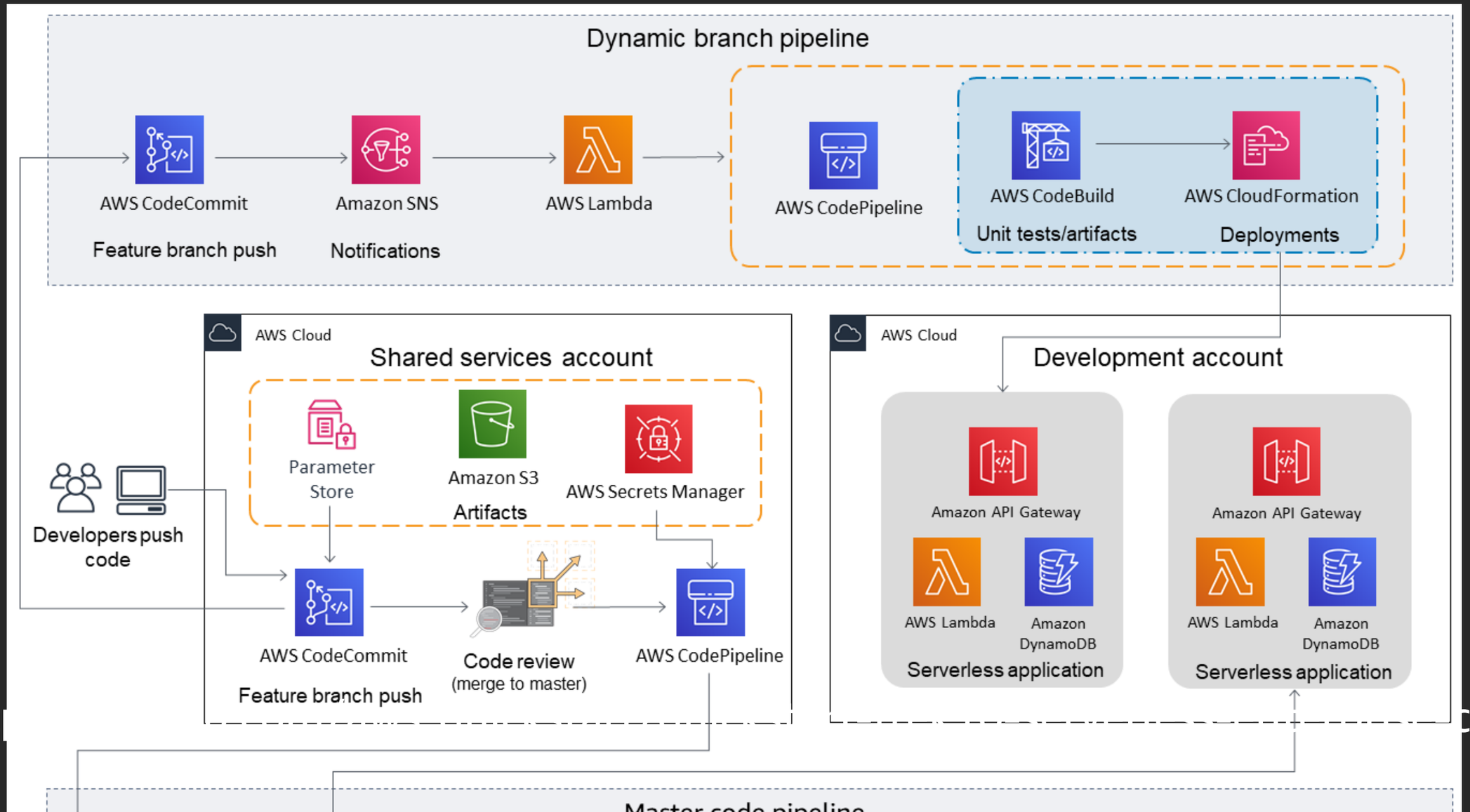


Phase 5

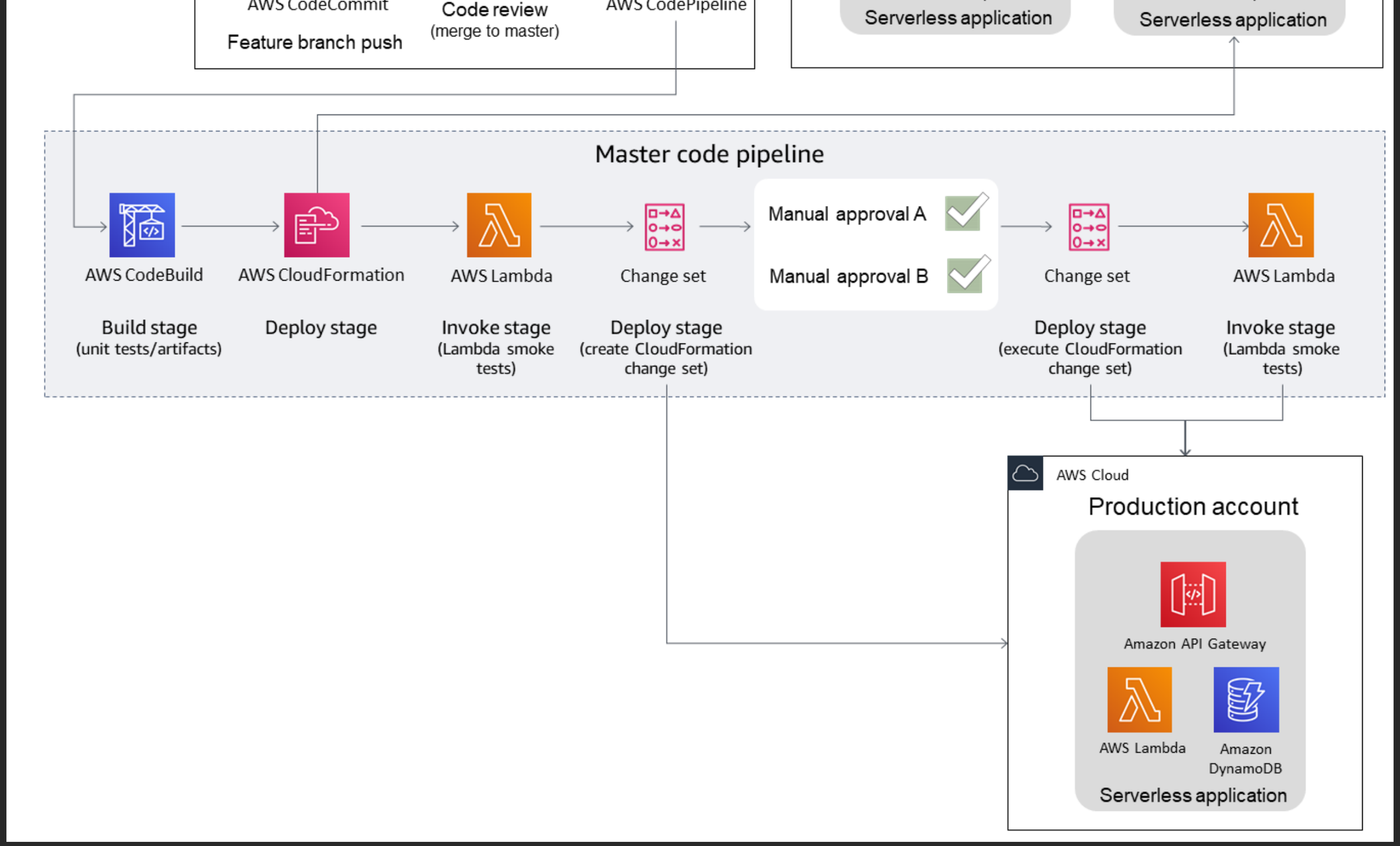
Continuous deployment



Complex pipeline example—Trek10



http cicd



https://github.com/aws-quickstart/quickstart-trek10-serverless-enterprise-cicd

IaC also for dev infrastructure

```
// 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'
});
```

IaC also for dev infrastructure

```
// create the build action
const buildProject = new codebuild.PipelineProject(pipelineStack, 'CodeBuildProje
ct', {
  projectName: 'DockerBuild',
  buildSpec: BuildSpec.fromSourceFilename('nginx/buildspec.yml'),
  environment: {
      buildimage: codebuild.LinuxBuildimage.STANDARD_2_0,
      privileged: true
  }
});
```

IaC also for dev infrastructure

```
// add codebuild permissions to access ECR (to push the image to the repo)
const role = <Role>buildProject.role;
role.addManagedPolicy(ManagedPolicy.fromAwsManagedPolicyName('AmazonEC2ContainerR
egistryPowerUser'));

const buildOutput = new pipeline.Artifact();
const buildAction = new pipeline_actions.CodeBuildAction({
    actionName: 'CodeBuildDockerimage',
    project: buildProject,
    input: sourceOutput,
    outputs: [buildOutput]
});
```

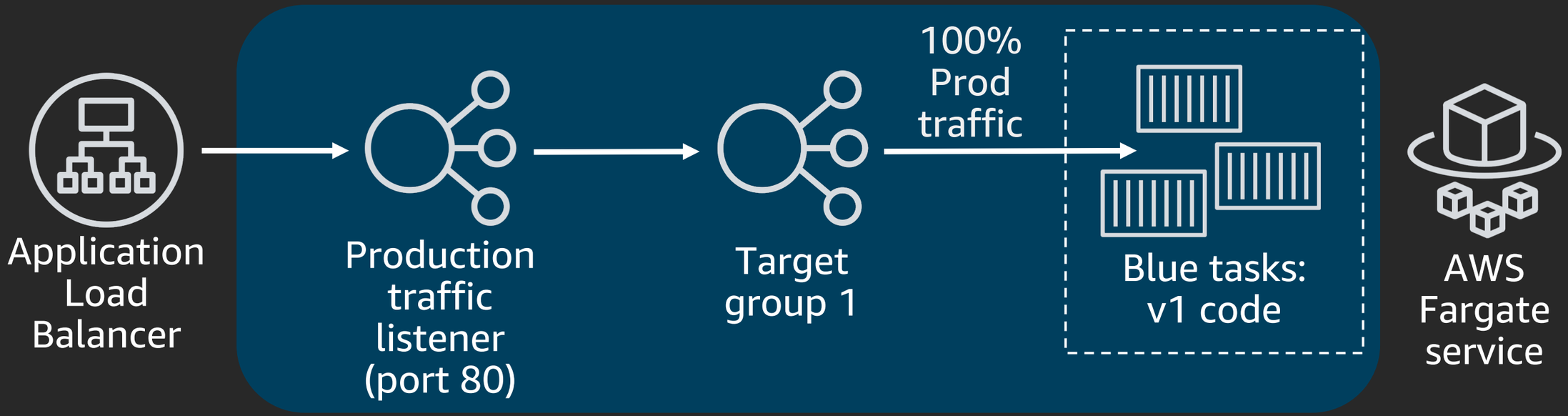
IaC also for dev infrastructure

```
const deployAction = new irEcsDeployAction({
            actionName: 'Deploy',
            serviceName: ecs.serviceName,
            clusterName: ecs.clusterName,
            input: buildOutput,
        });
```

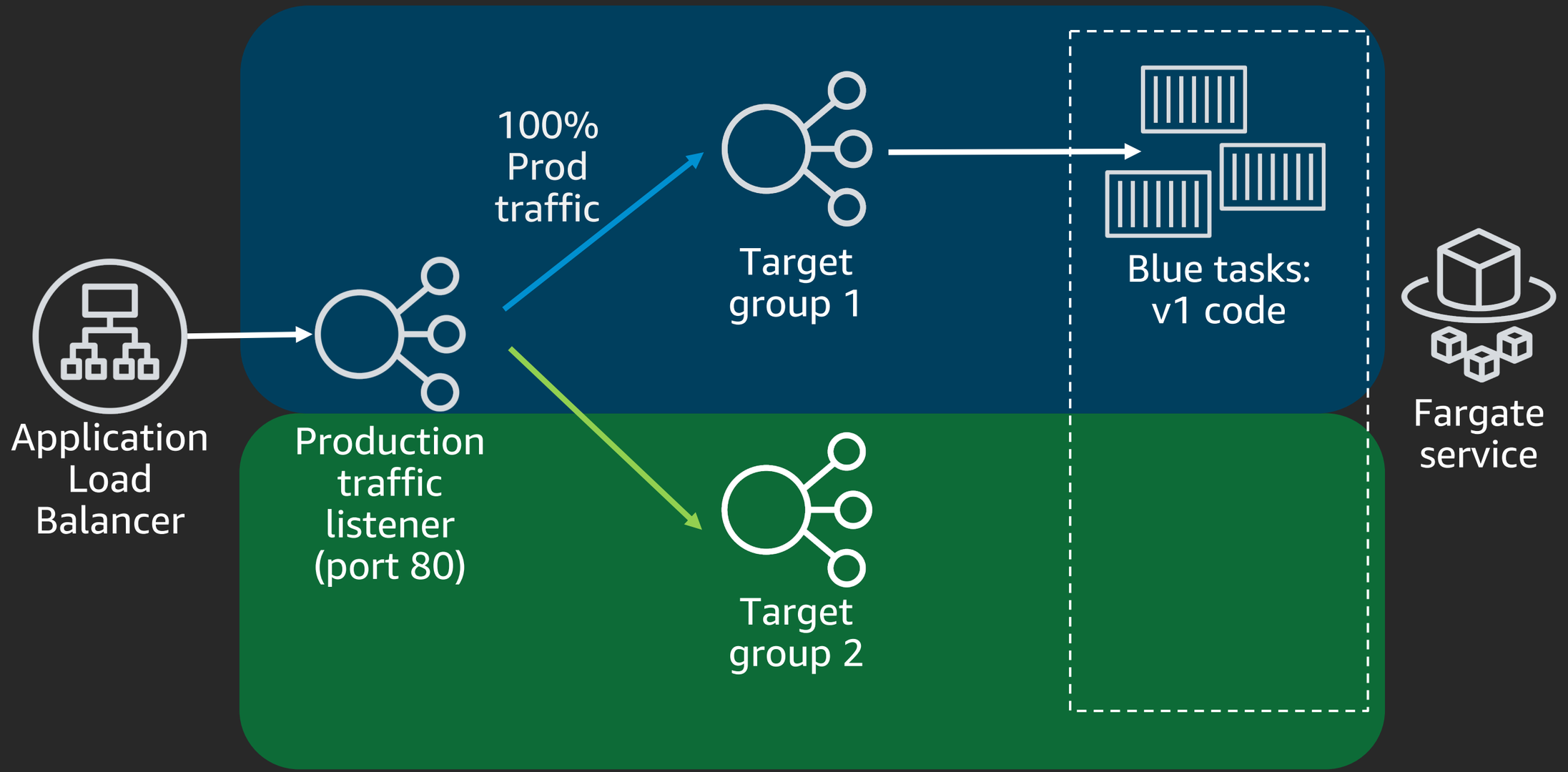
IaC also for dev infrastructure

```
// finally, create the pipeline
const codePipeline = new pipeline.Pipeline(pipelineStack, 'Pipeline', {
    pipelineName: 'ECSDeploy',
    stages: [
      {
        stageName: 'GetSource',
        actions: [sourceAction],
      },
      {
        stageName: 'BuildDockerimage',
        actions: [buildAction]
      },
      {
        stageName: 'DeployToEcs',
        actions: [deployAction]
      }
    ],
  });
```

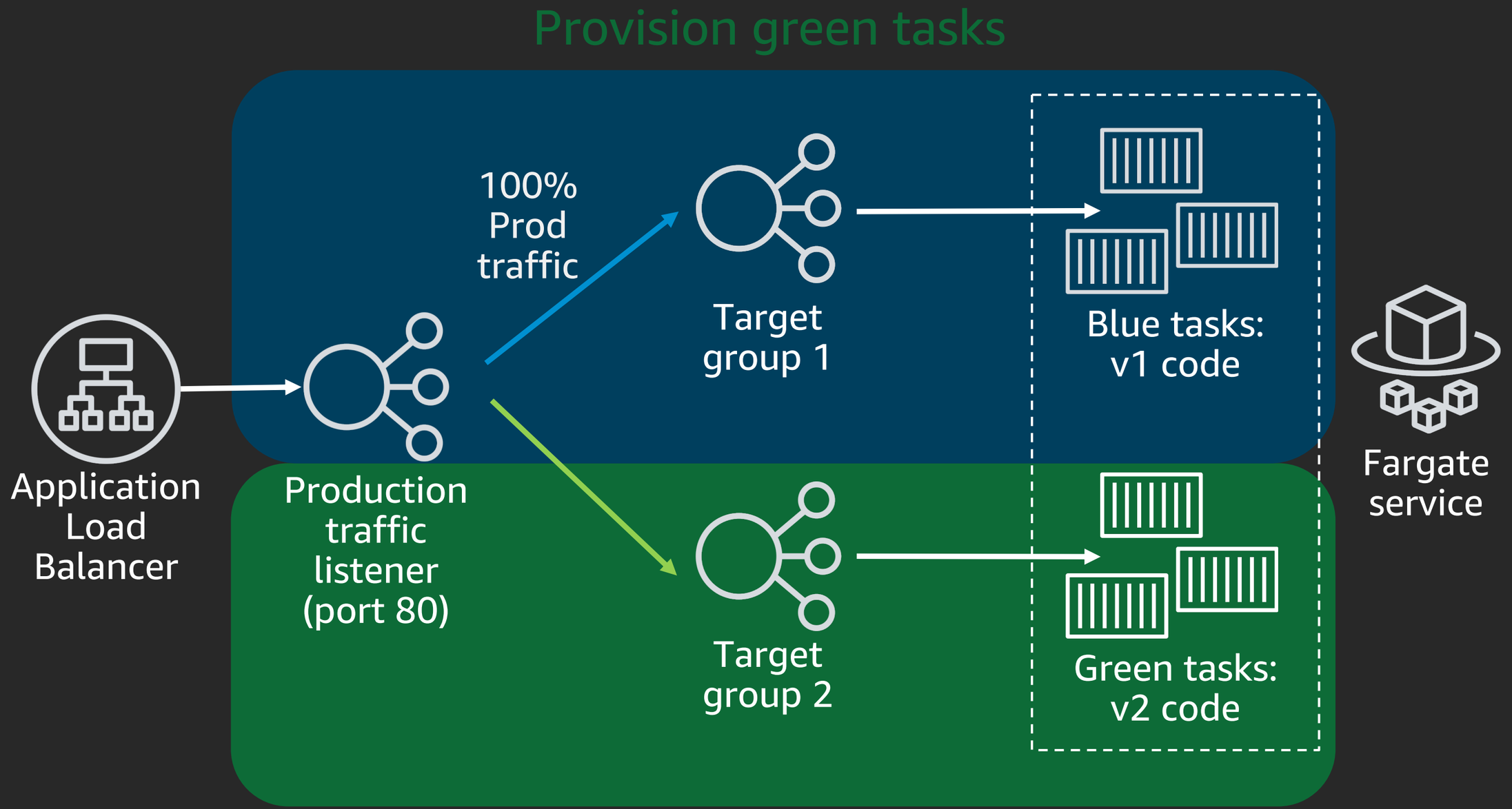
Blue-green deployment



Blue-green deployment

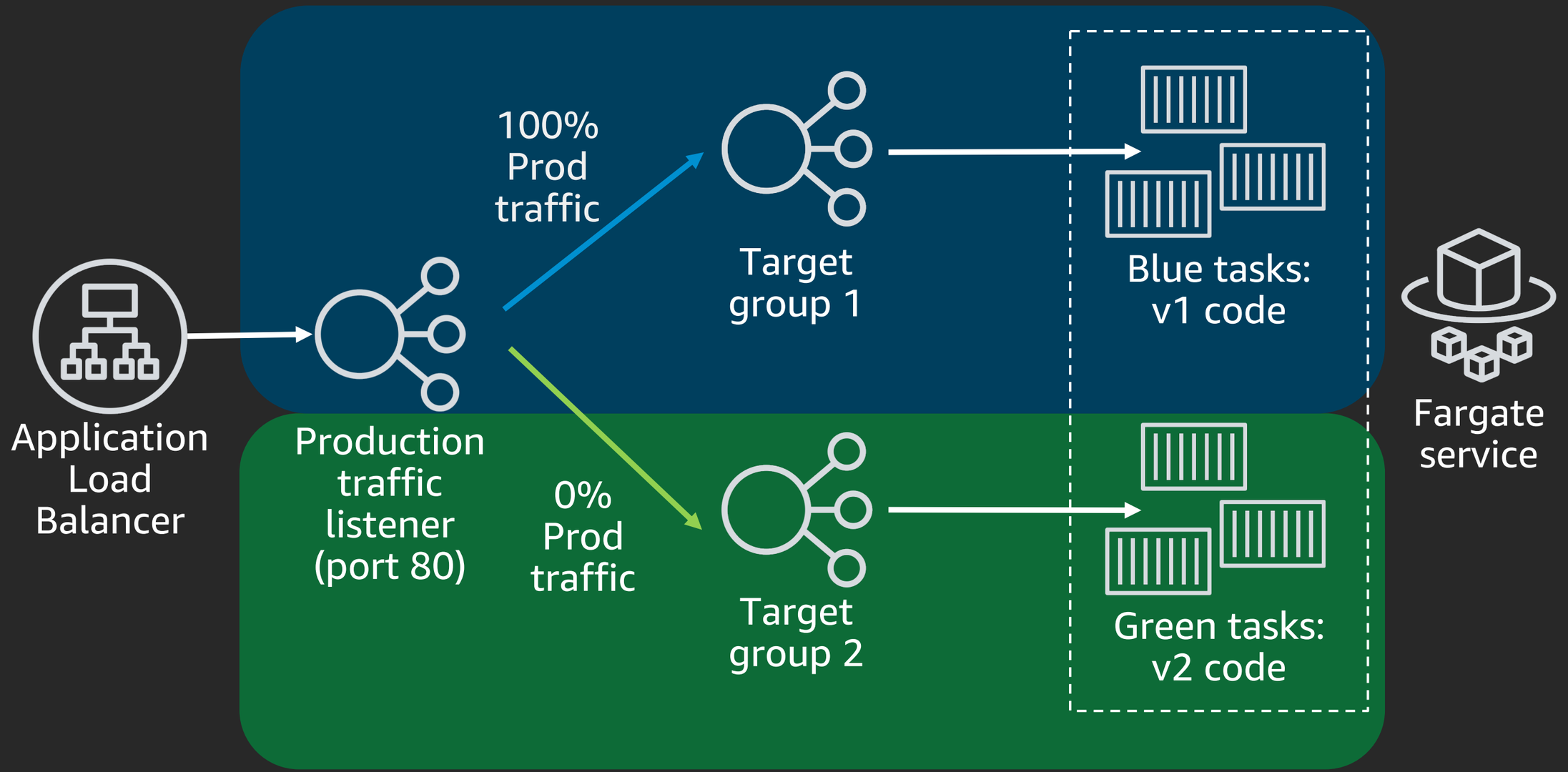


Blue-green deployment



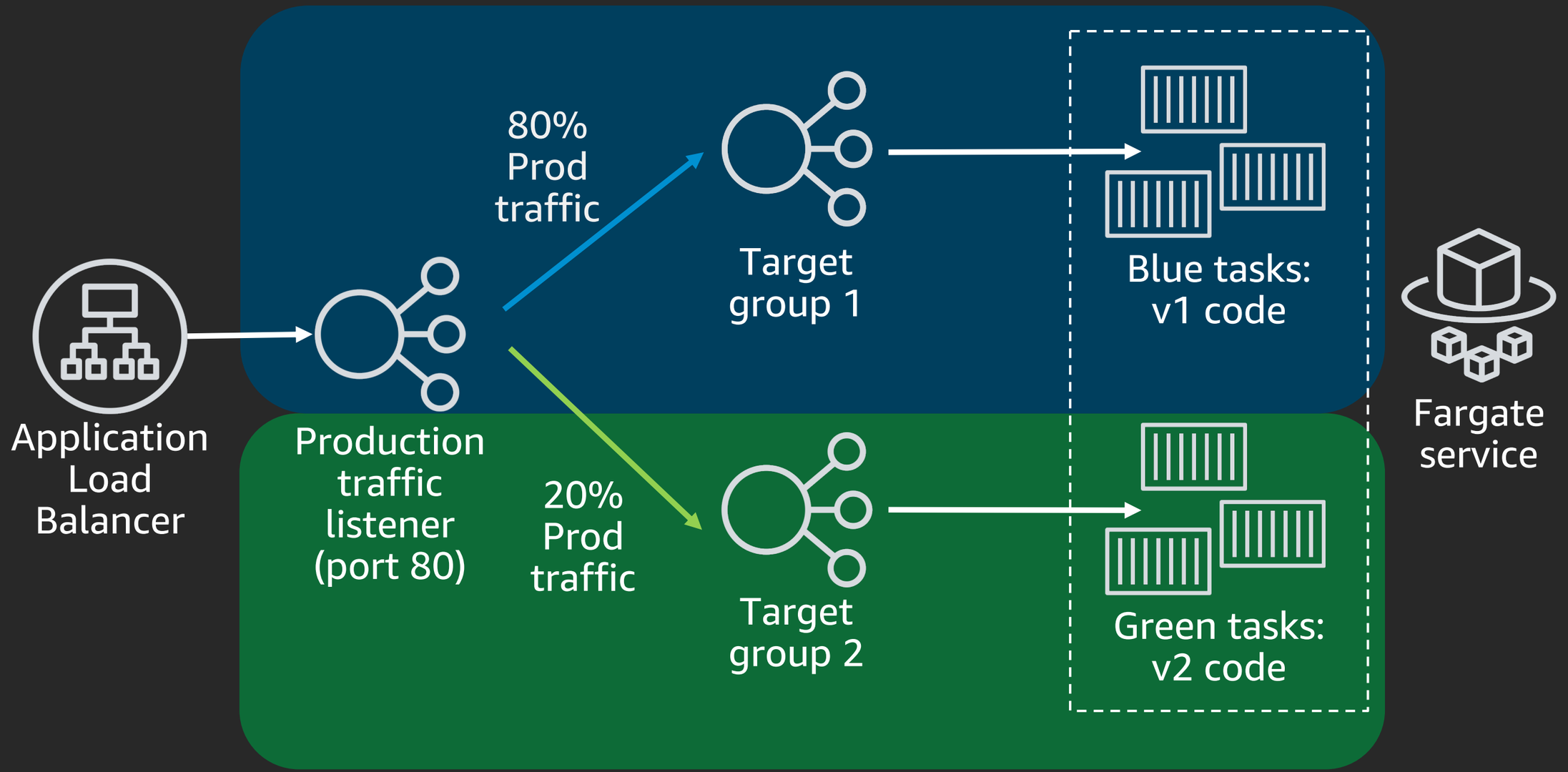
Blue-green deployment

Run hook against test endpoint before green tasks receive prod traffic

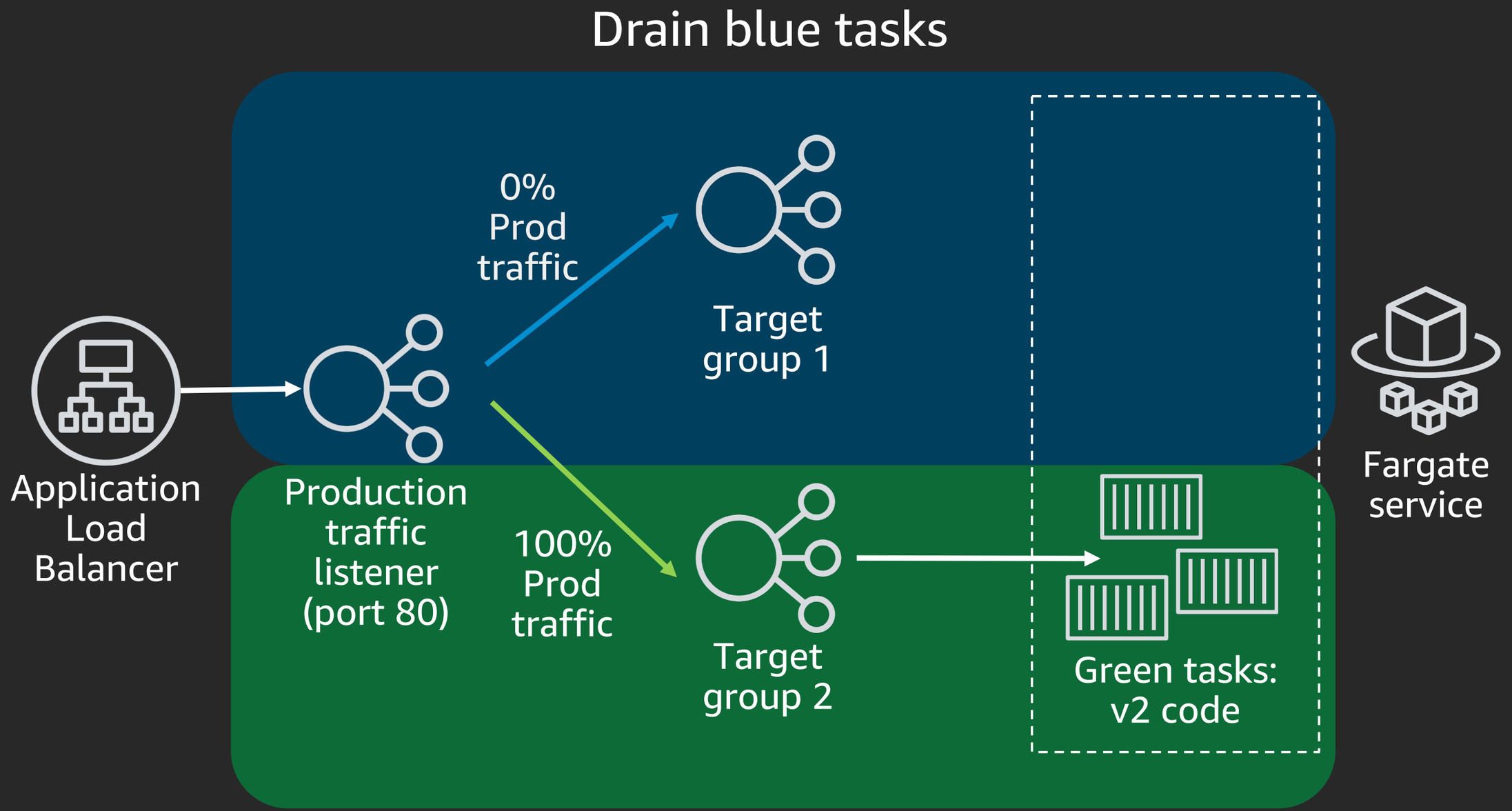


Blue-green deployment

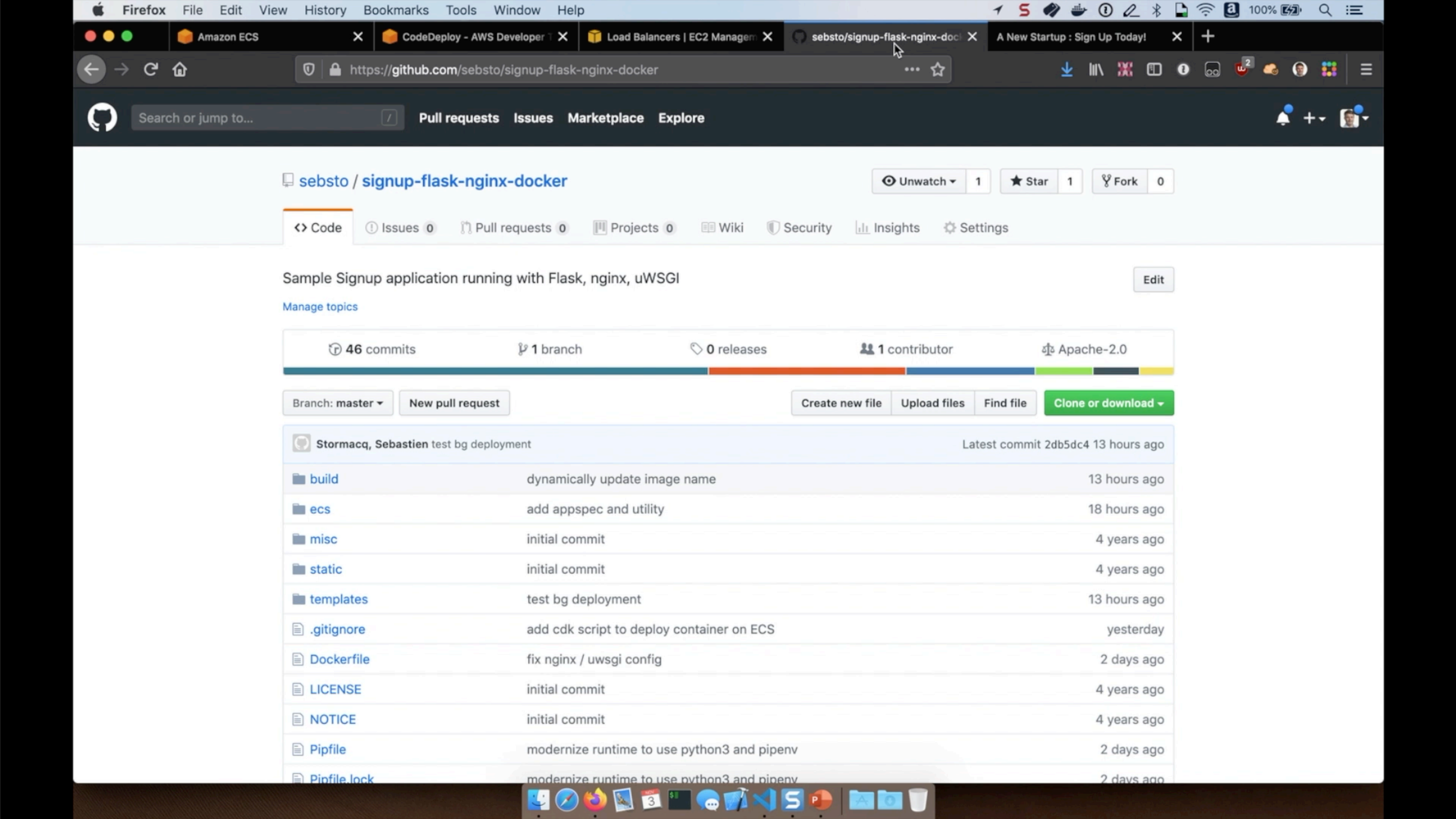
Flip traffic to green tasks, rollback in case of alarm



Blue-green deployment

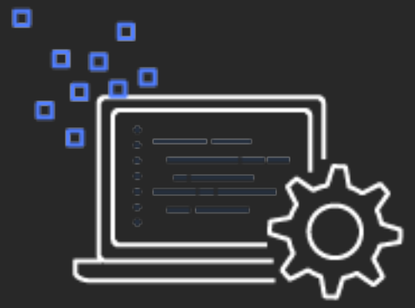


Demo

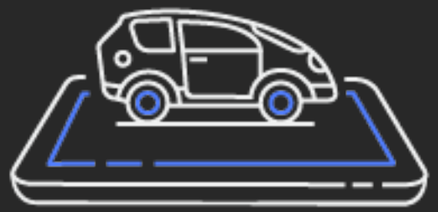

Firefox File Edit View History Bookmarks Tools Window Help
Amazon ECS
CodeDeploy - AWS Developer
Load Balancers | EC2 Manage
sebsto/signup-flask-nginx-doc
A New Startup : Sign Up Today!
https://github.com/sebsto/signup-flask-nginx-docker
Search or jump to...
Pull requests
Issues
Marketplace
Explore
sebsto / signup-flask-nginx-docker
Unwatch 1
Star 1
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings
Sample Signup application running with Flask, nginx, uWSGI
Edit
Manage topics
46 commits
1 branch
0 releases
1 contributor
Apache-2.0
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
Stormacq, Sebastien test bg deployment
Latest commit 2db5dc4 13 hours ago
build dynamically update image name 13 hours ago
ecs add appspec and utility 18 hours ago
misc initial commit 4 years ago
static initial commit 4 years ago
templates test bg deployment 13 hours ago
.gitignore add cdk script to deploy container on ECS yesterday
Dockerfile fix nginx / uwsgi config 2 days ago
LICENSE initial commit 4 years ago
NOTICE initial commit 4 years ago
Pipfile modernize runtime to use python3 and pipenv 2 days ago
Pipfile.lock modernize runtime to use python3 and pipenv 2 days ago

Value-added capabilities debugging

Debugging modern apps locally is hard



Applications
are large



Mockups
are not perfect

New
developer patterns

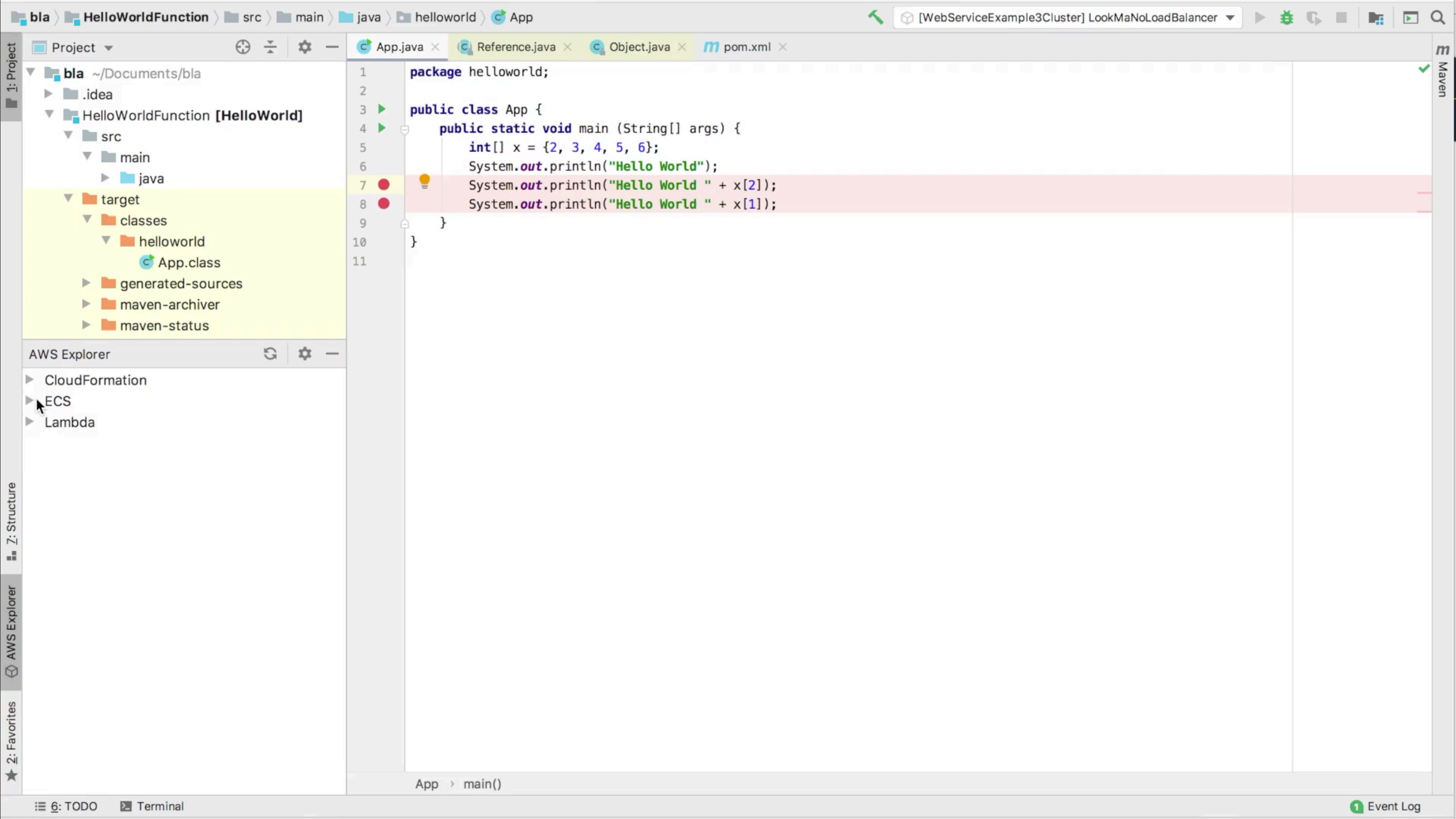
You are debugging in the cloud


```
1928        # request came with the OPTIONS method, reply automatically
1929        if (
1930            getattr(rule, "provide_automatic_options", False)
1931            and req.method == "OPTIONS"
1932        ):
1933            return self.make_default_options_response()
1934        # otherwise dispatch to the handler for that endpoint
1935        return self.view_functions[rule.endpoint](**req.view_args)
1936
1937    def full_dispatch_request(self):
1938        """Dispatches the request and on top of that performs request
1939        pre and postprocessing as well as HTTP exception catching and
1940        error handling.
1941
1942        .. versionadded:: 0.7
1943        """
1944        self.try_trigger_before_first_request_functions()
1945        try:
1946            request_started.send(self)
1947            rv = self.preprocess_request()
1948            if rv is None:
```

aws AWS Cloud

Amazon Elastic Container Service

Demo

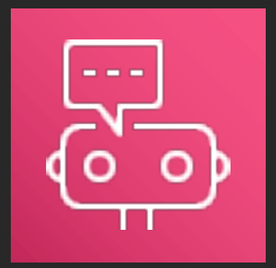

bla
HelloWorldFunction
src
main
java
helloworld
App
[WebServiceExample3Cluster] LookMaNoLoadBalancer
Project
bla ~/Documents/bla
.idea
HelloWorldFunction [HelloWorld]
src
main
java
target
classes
helloworld
App.class
generated-sources
maven-archiver
maven-status
AWS Explorer
CloudFormation
ECS
Lambda
App.java
Reference.java
Object.java
pom.xml
1 package helloworld;
2
3 public class App {
4 public static void main (String[] args) {
5 int[] x = {2, 3, 4, 5, 6};
6 System.out.println("Hello World");
7 System.out.println("Hello World " + x[2]);
8 System.out.println("Hello World " + x[1]);
9 }
10 }
11
App › main()
1: Project
7: Structure
AWS Explorer
2: Favorites
Maven
6: TODO
Terminal
Event Log

Value-added capabilities
ChatOps

ChatOps

AWS Chatbot

Events



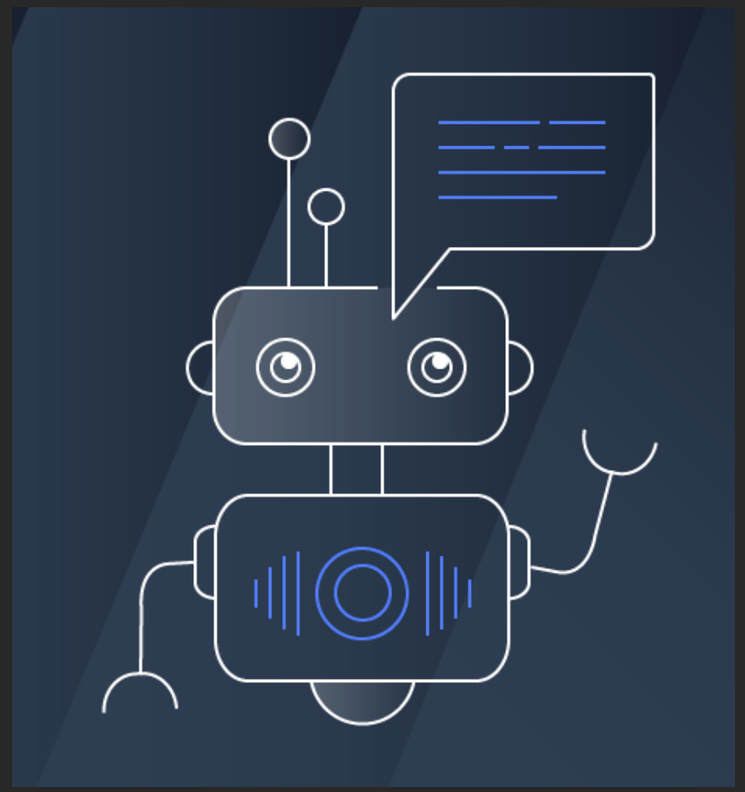
AWS Chatbot

Receive notifications from AWS services about infrastructure events, billing, security, and more

Easily integrate with Slack

Built-in security templates for common use cases simplify configuration and enable best practices

AWS Chatbot can now run commands



AWS Chatbot

Interactive agent for ChatOps

- Receive notifications
- Run commands for diagnostic information
- Pre-defined IAM policy templates
- Support for Slack and Chime

Demo

Firefox File Edit View History Bookmarks Tools Window Help

Lambda Management Console | CloudWatch Management Cons | Chatbot Console | Slack | chatbot | aws-sst-demo

https://app.slack.com/client/TM3LYNKMX/CPSTRSREX/details/apps

aws-sst-demo
Seb

Jump to...

Threads
Apps

Channels
chatbot
general
lex-demo
random

+ Add a channel

Direct Messages
Slackbot
Seb (you)

+ Invite people

Recent Apps
aws
lexdemo
+ Add apps

Upgrade

#chatbot

1 | 0 | Add a topic

Search

Bring your team into Slack

Slack is better with teammates – invite them to start collaborating.

Add People

Seb 11:00 AM
test message

¯_(ツ)_/¯

Message #chatbot

B I S </> Aa @

About this channel

Channel Details
Highlights
Pinned Items
1 Member
1 App

aws APP

Add App

Shared Files
Notification Preferences

How Amazon is doing it

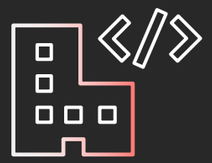
How Amazon does DevOps?

Decompose for agility
(microservices, 2 pizza teams)

Automate everything

Standardized tools

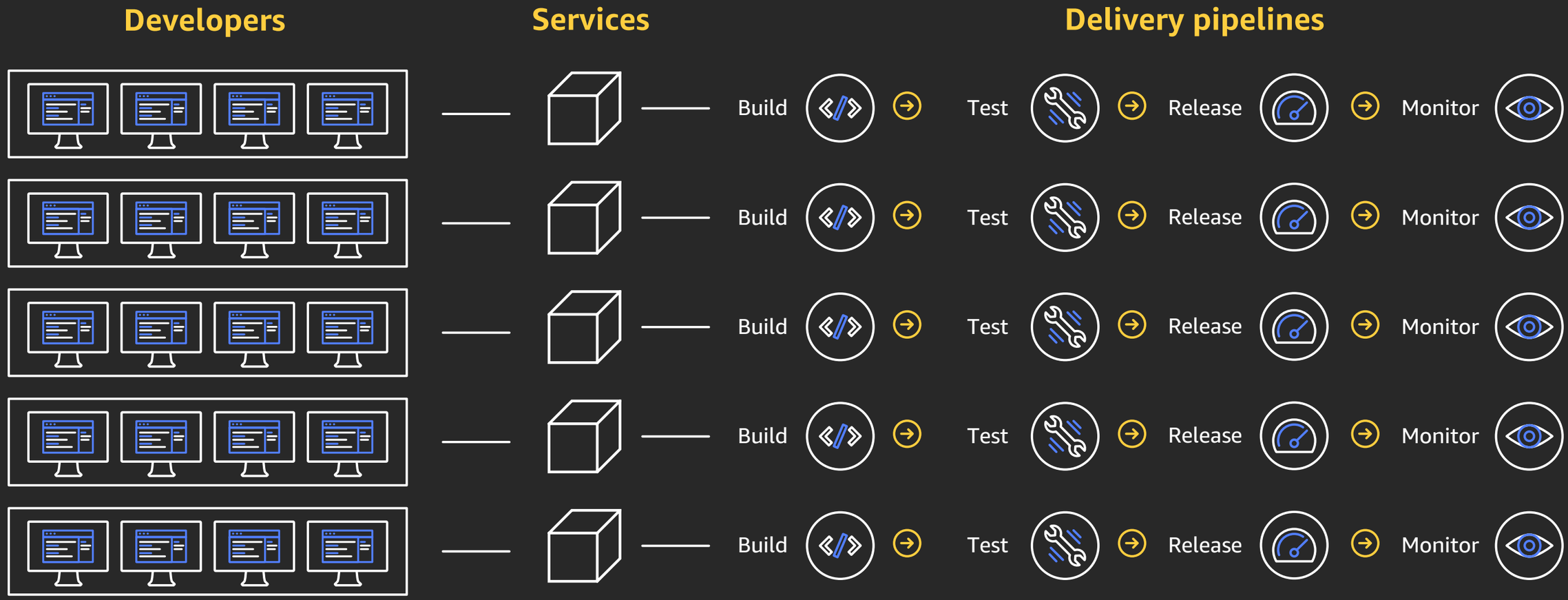
Belts and suspenders
(governance, templates)



infrastructure as code

Wrap-up

What we built



Trunk-based source code control

AWS CDK

Think big—impossibly big

Start small

Iterate

Start anywhere

but

start somewhere

Thank you!

Sébastien Stormacq

@sebsto

Jonathan Weiss

@jweiss

Please complete the session survey in the mobile app.