Serverless CI/CD pipelines for your DevOps needs

@dragospm

Dragos Madarasan: Solutions Architect

September 19, 2019

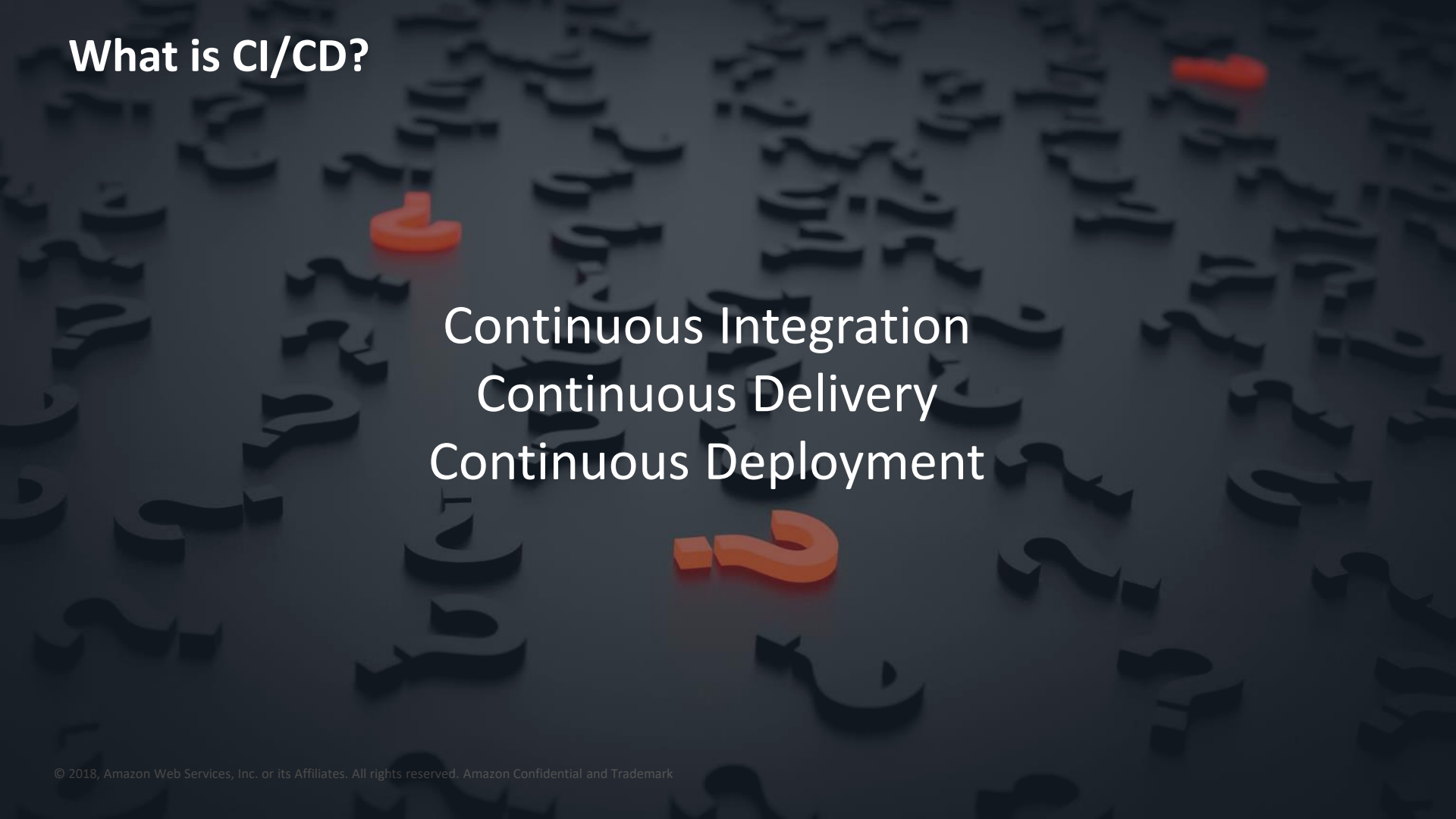
Who am I?

- Solutions Architect @ AWS, covering Romania and Hungary
- Work with and support clients, partners, NGOs, Public Sector
- Based in Munich, Germany, often in Romania
- Previously Support Engineer and ProServe Consultant @ AWS
- Used to manage a WordPress blog, now I host it on Amazon S3

What is CI/CD?

And why is it important

What is CI/CD?



Continuous Integration
Continuous Delivery
Continuous Deployment

The Pipeline

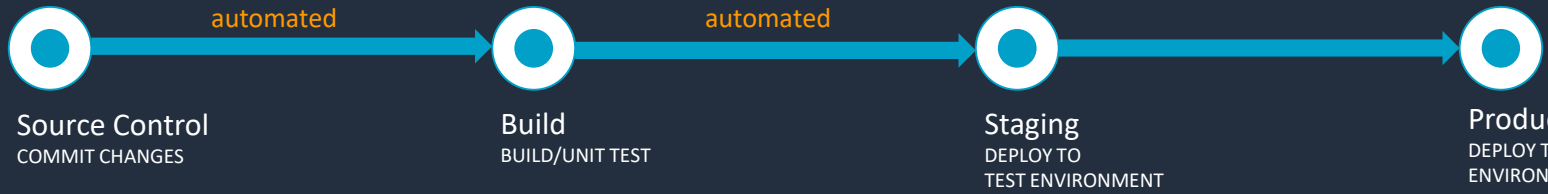
Source Control
COMMIT CHANGES

Build
BUILD/UNIT TEST

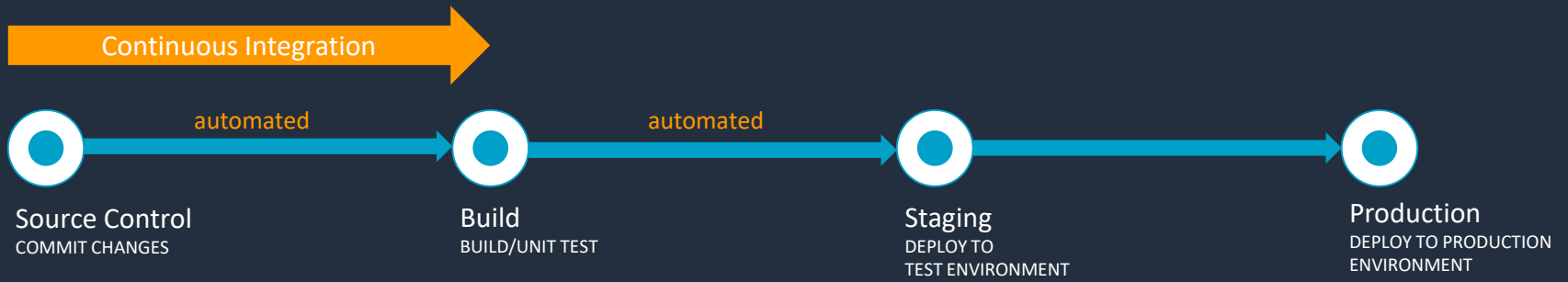
Staging
DEPLOY TO
TEST ENVIRONMENT

Production
DEPLOY TO PRODUCTION
ENVIRONMENT

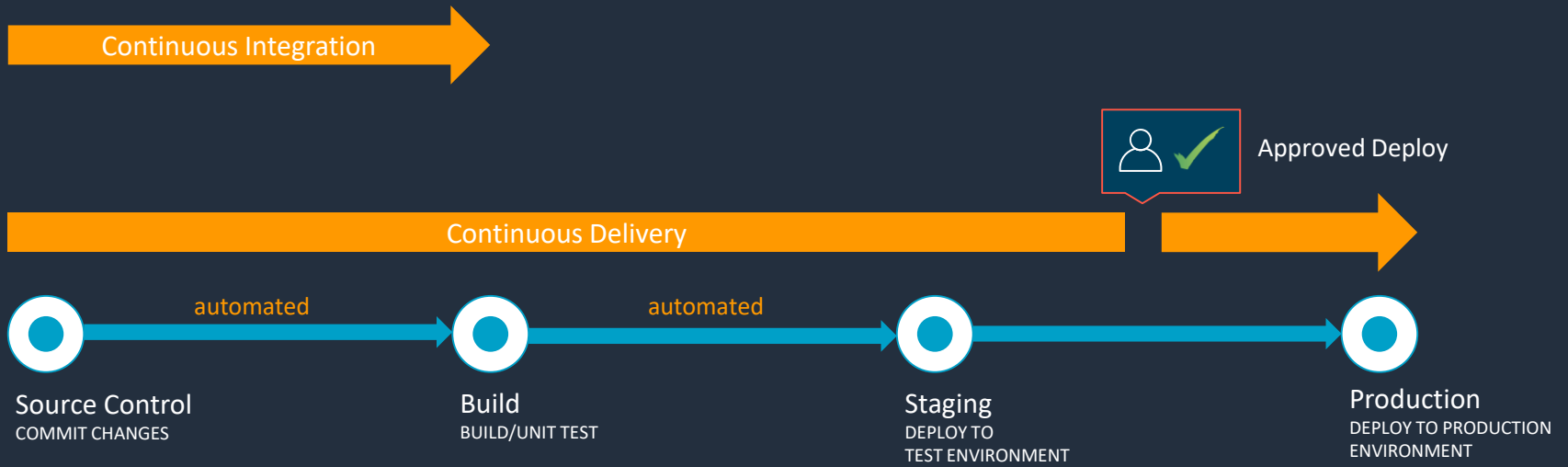
The Automated Pipeline



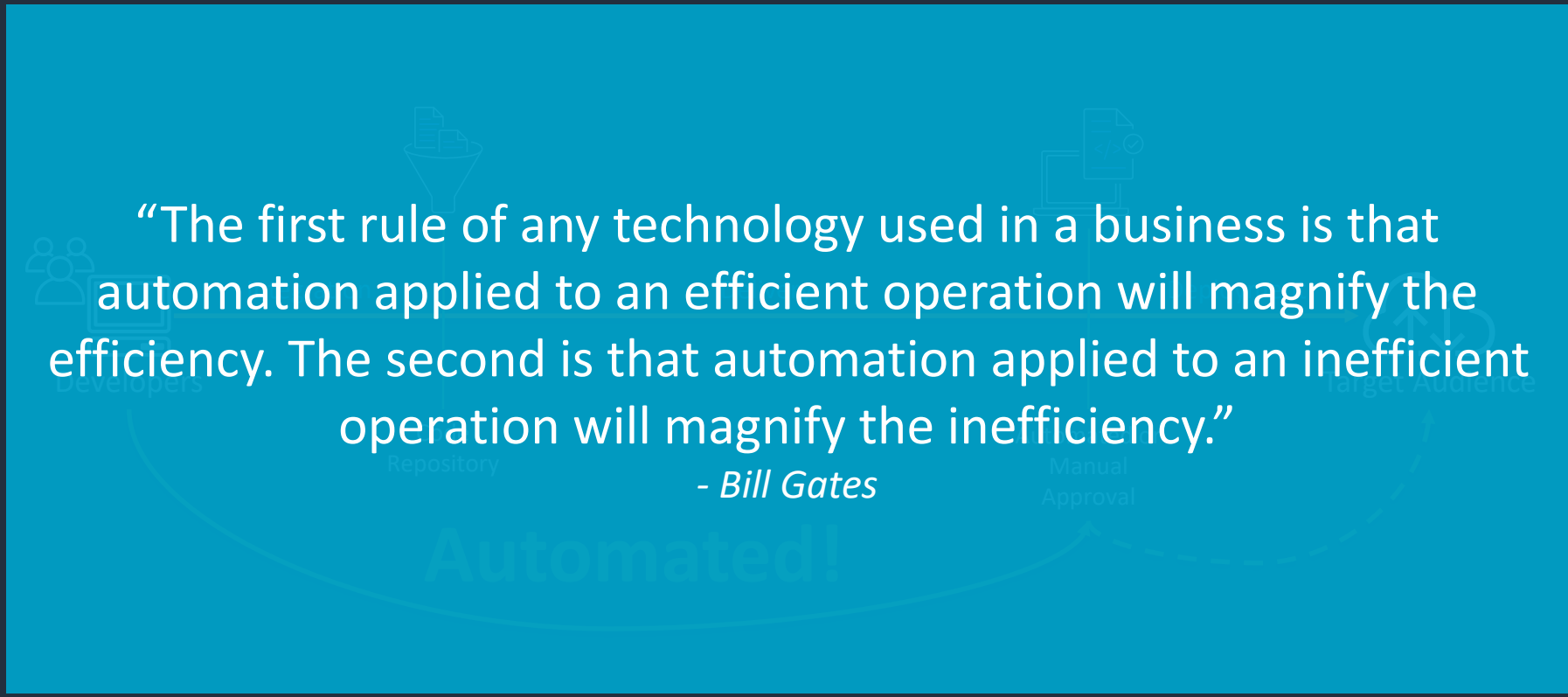
The Pipeline – Continuous Integration



The Pipeline – Continuous Delivery



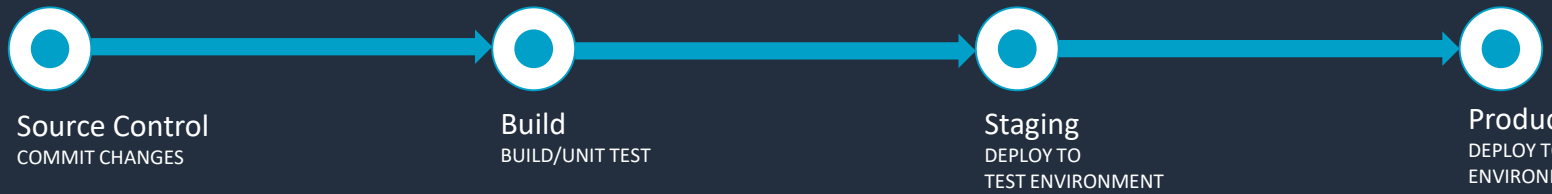
What is CI/CD?



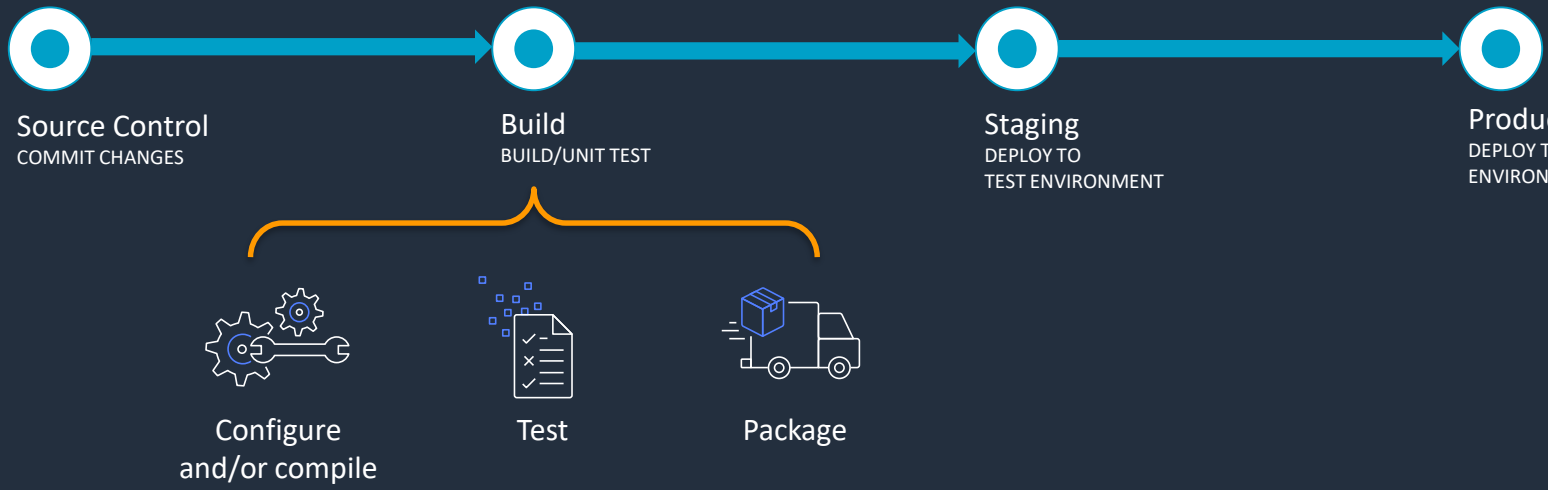
"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency."

- Bill Gates

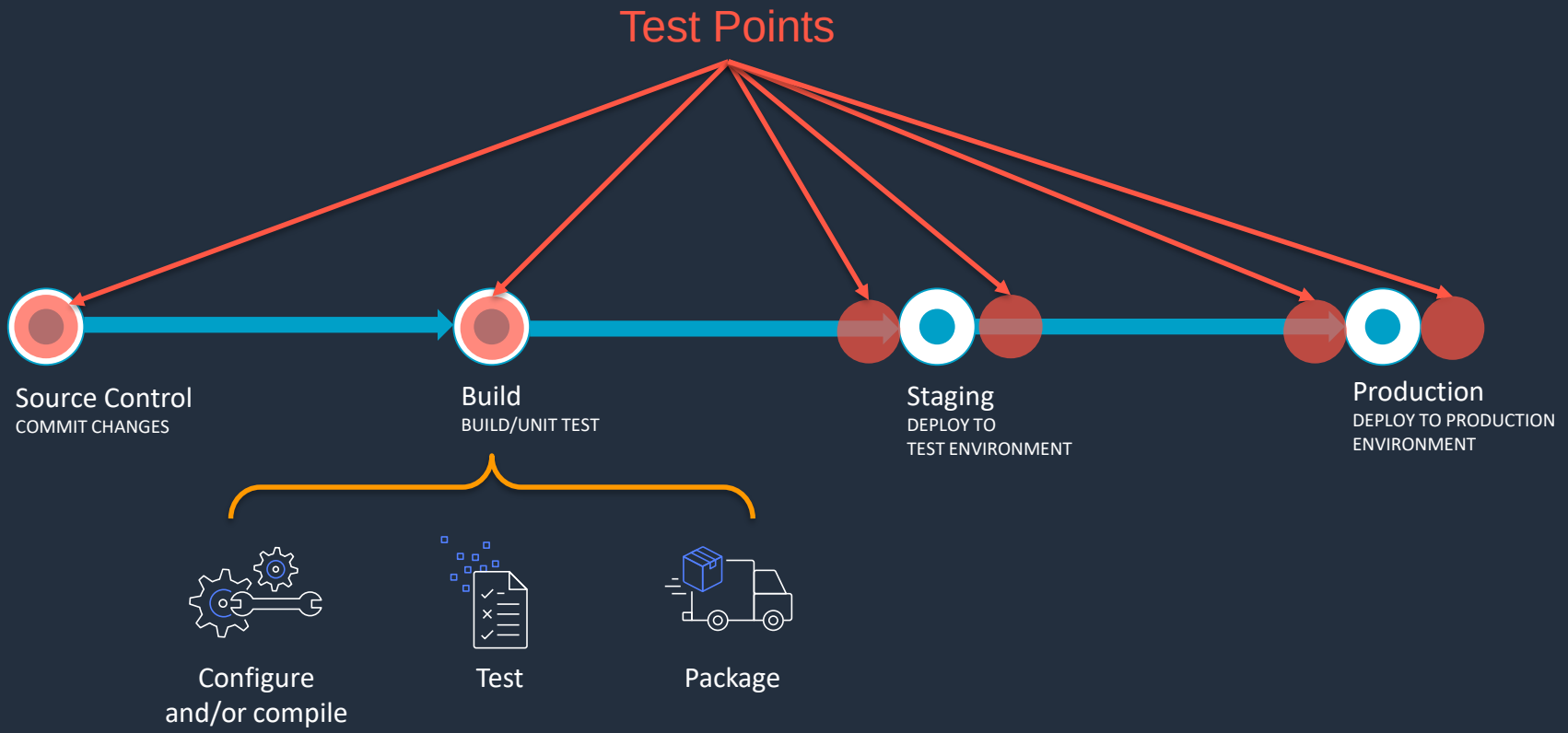
The Pipeline



The Pipeline



The Pipeline



Code Repository

AWS CodeCommit

Fully-managed source control service that hosts secure Git-based repositories

Allows teams to collaborate on code in a secure and highly scalable ecosystem

Automatically encrypts your files in transit and at rest

Integrated with AWS Identity and Access Management (IAM)

https://aws.amazon.com/codecommit/

Third Party Code Repositories

GitHub

Atlassian Bitbucket

GitLab

git private repo

Integrates with CodeBuild and CodePipeline

Integrates with CodeBuild

https://secure.flickr.com/photos/spenceyc/7481166880

AWS CodeBuild

Fully-managed build service that can compile source code, run tests, and produce software packages

Scales continuously and processes multiple builds concurrently

Can consume environment variables from AWS SSM Parameter Store

Can run in your VPC and locally

Supports dependency caching

https://aws.amazon.com/codebuild/

buildspec.yml Example

```
version: 0.1

environment_variables:
  plaintext:
      "INPUT_FILE": "saml.yaml”
      "S3_BUCKET": ""
phases:
  install:
    commands:
      - npm install
pre_build:
    commands:
      - eslint *.js
build:
    commands:
      - npm test
post_build:
    commands:
      - aws cloudformation package --template $INPUT_FILE --s3-
bucket $S3_BUCKET --output-template post-saml.yaml
artifacts:
  type: zip
  files:
      - post-saml.yaml
      - beta.json
```

buildspec.yml Example

```
version: 0.1

environment_variables:
  plaintext:
      "INPUT_FILE": "saml.yaml”
      "S3_BUCKET": ""
phases:
  install:
    commands:
      - npm install
pre_build:
    commands:
      - eslint *.js
build:
    commands:
      - npm test
post_build:
    commands:
      - aws cloudformation package --template $INPUT_FILE --s3-
bucket $S3_BUCKET --output-template post-saml.yaml
artifacts:
  type: zip
  files:
      - post-saml.yaml
      - beta.json
```

- Variables to be used by phases of build
- Examples for what you can do in the phases of a build:
  - You can install packages or run commands to prepare your environment in ”install”.
  - Run syntax checking, commands in “pre_build”.
  - Execute your build tool/command in “build”
  - Test your app further or ship a container image to a repository in post_build
- Create and store an artifact in Amazon S3

Build Specification – Phases

Phase	Description	Examples
install	Installation of packages into the environment	Install testing frameworks e.g. RSpec, Mocha
pre_build	Commands to run before the build such as login steps or installation of dependencies	Log in to Amazon ECR. run Ruby bundler or npm
build	Sequence to run the build such as compilation and/or running tests	Run go build, sbt, Mocha, RSpec
post_build	Commands to run after a build on success or failure	Build a JAR via Maven or push a Docker image to Amazon ECR

Best Practices

- Tag output artifacts to source control revisions (e.g. git SHA, semantic version)
- Avoid using a “latest” or “production” tag
- Optimize for build speed
- Collocate build process with its artifact repository

https://www.flickr.com/photos/seattlemunicipalarchives/12504672623/

AWS CodePipeline

Continuous delivery service for fast and reliable application updates

Model and visualize your software release process

Builds, tests, and deploys your code every time there is a code change

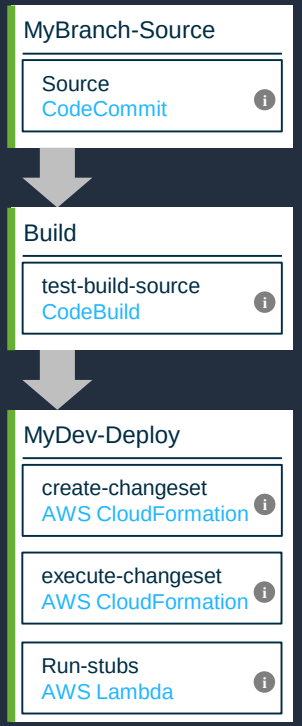
Integrates with third-party tools and AWS

https://aws.amazon.com/codepipeline/

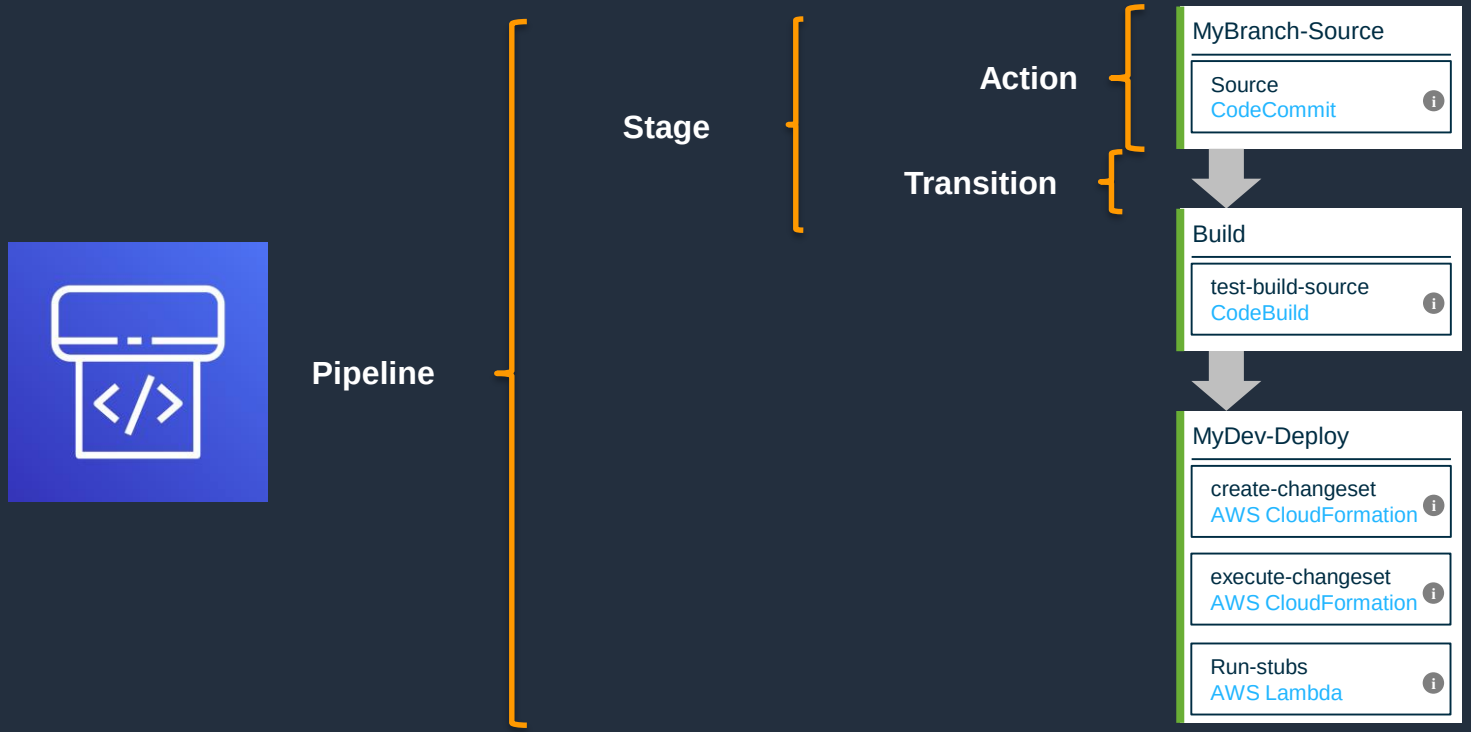
Example of minimal developer's pipeline

This pipeline:

- Three Stages
- Builds code artifact
- One Development environment
- Uses SAM/CloudFormation to deploy artifact and other AWS resources
- Has Lambda custom actions for running my own testing functions



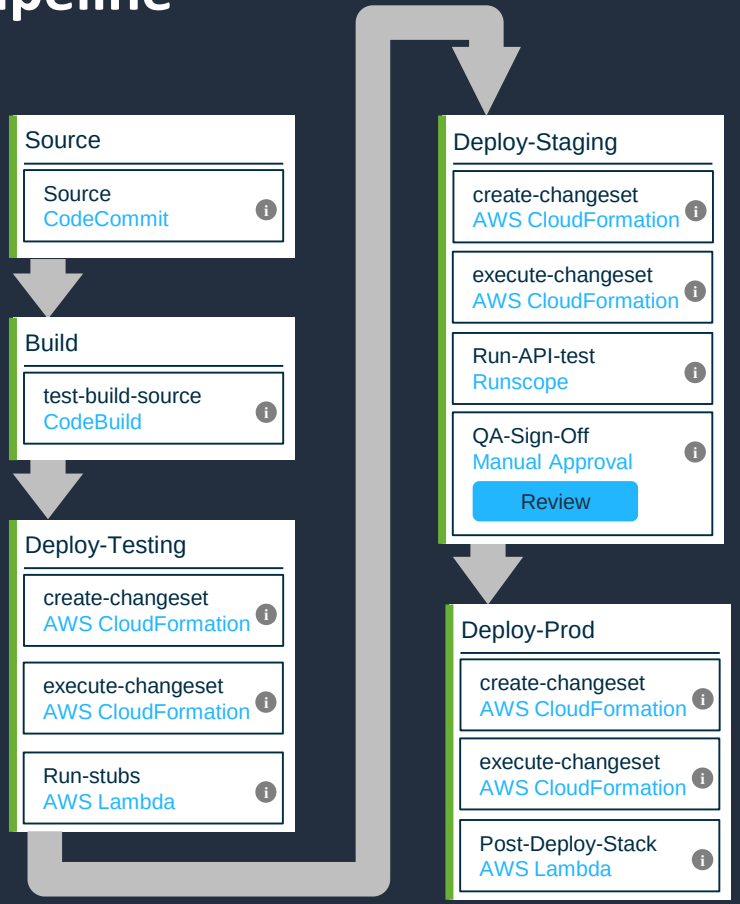
Example of minimal developer's pipeline



Example of minimal Developer's pipeline

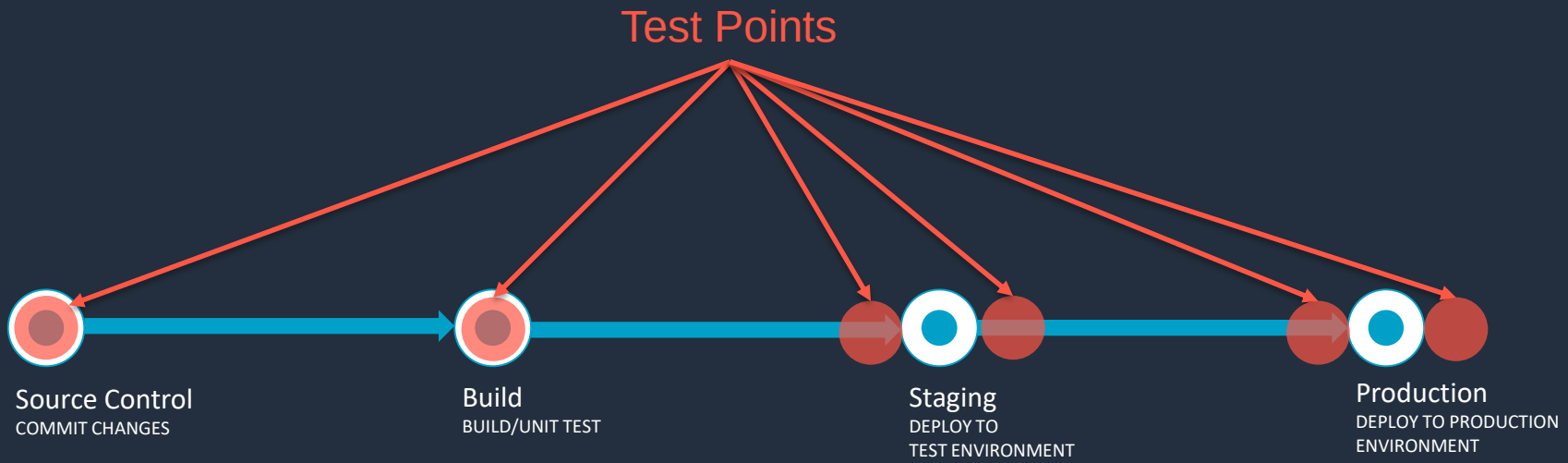
This pipeline:

- Five Stages
- Builds code artifact
- Three deployed to "Environments"
- Uses SAM/CloudFormation to deploy artifact and other AWS resources
- Has Lambda custom actions for running my own testing functions
- Integrates with a 3rd party tool/service
- Has a manual approval before deploying to production

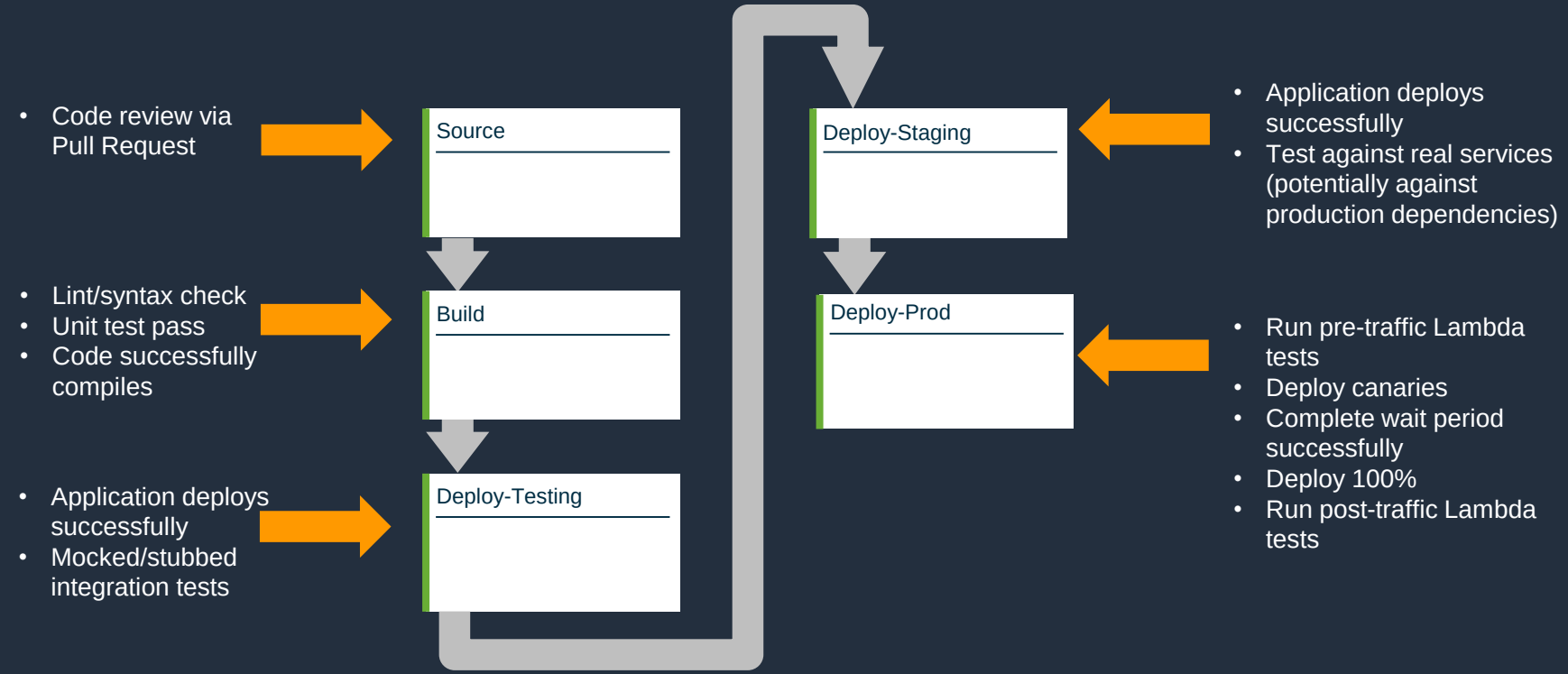


Testing

The Pipeline – Continuous Deployment



Where and what to test



AWS CloudFormation

AWS CloudFormation

Provides a common language for you to describe and provision all the infrastructure resources in your cloud environment

Build and rebuild your infrastructure and applications, without having to perform manual actions or write custom scripts.

https://aws.amazon.com/cloudformation/

Jekyll

- **Static site generator**
- **Written in Ruby**
- **Blog-aware**
- **Custom theme support**
- **Works with Apache, Nginx, GitHub Pages and Amazon S3**

https://jekyllrb.com/

Demo

AWS Webinar Series

Ep. 1 – AWS Cloud Concepts: http://bit.ly/AwsWebinarRO1

Ep. 2 – AWS Technology: http://bit.ly/AwsWebinarRO2

Ep. 3 – Security on AWS: October 2019

Ep. 4 – Architecting on AWS: October 2019

Ep. 5 – AWS Pricing and Support: November 2019

AWSome Day in Iasi!

Date: Tuesday, October 22, 2019

Time: 9:30 am – 4:30 pm,
Registration at: 8:30 am

Location: Congress Hall, Sala Rossini,

https://pages.awscloud.com/awsome-day-iasi_2019.html

https://pages.awscloud.com/awsome-day-iasi_2019.html

AWS Popup Loft Bucharest

Date: November 11-15, 2019

Time: 9:00 – 5:30

Location: Impact Hub Timpuri Noi, Bucharest

https://awsloft-bucharest.com (not live yet)

Q&A

How can I help?

Dragos Madarasan – Solutions Architect
madarasa@amazon.com