Builders Online Series

Getting Started with DevOps on AWS

Loh, Yiang Meng

Solutions Architect,
Amazon Web Services

Agenda

- What is DevOps?
- Release Process
- What is CI/CD?
- AWS Code Services
- 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

- Leadership
- Organization
- Individuals

- Architecture patterns
- Coding practice
- Code packaging
- Service discovery
- Secrets management

- Small, frequent updates
- Microservice patterns
- Caching and object access patterns
- Circuit breaker
- Code promotions

aws

What is DevOps?

DevOps = Culture + Practices + Tools

Culture	Practices		Tools	
Leadership	Architecture patterns	Small, frequent updates	Source code	Logging
Organization	Coding practice	Microservice patterns	Continuous integration	Monitoring
Individuals	Code packaging	Caching and object access patterns	Continuous delivery	Alerting
	Service discovery	Circuit breaker	Infrastructure as Code	Adaptive improvements
	Secrets management	Code promotions	Static analysis	Testing

aws

A typical release process stages

A typical release process stages

- Pull source code such as .java files

A typical release process stages

- Pull source code such as .java files

- Compile code
- Unit tests
- Style checkers
- Generate code artifacts

A typical release process stages

- Pull source code such as .java files

- Compile code
- Unit tests
- Style checkers
- Generate code artifacts

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

A typical release process stages

- Pull source code such as .java files

- Compile code
- Unit tests
- Style checkers
- Generate code artifacts

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

- Deploy to staging environments
- Promote to Production environment

Common issues faced

- Merge conflicts
- Tightly coupled changes
- Repository unavailability

Common issues faced

- Merge conflicts
- Tightly coupled changes
- Repository unavailability

- Inconsistent configuration
- Long wait for build server availability

Common issues faced

- Merge conflicts
- Tightly coupled changes
- Repository unavailability

- Inconsistent configuration
- Long wait for build server availability

- Simply going through motion
- Inconsistent testing
- Unable to do load testing

Common issues faced

- Merge conflicts
- Tightly coupled changes
- Repository unavailability

- Inconsistent configuration
- Long wait for build server availability

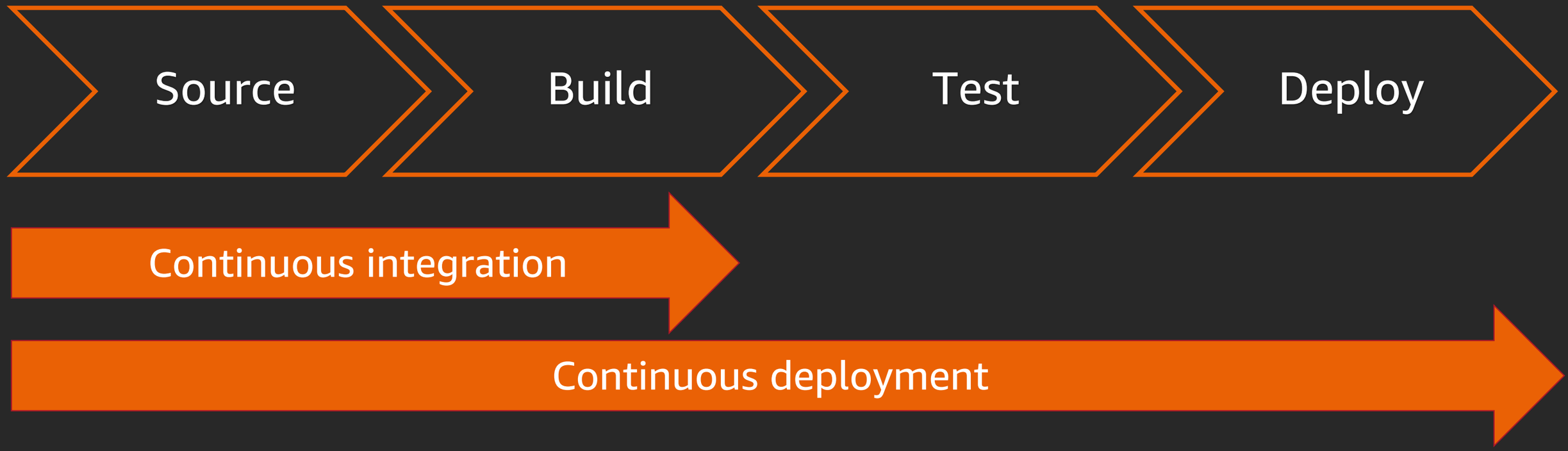
- Simply going through motion
- Inconsistent testing
- Unable to do load testing

- Deployment downtime
- “But it worked in Staging!”
- Inconsistent rollback

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

Continuous deployment goals

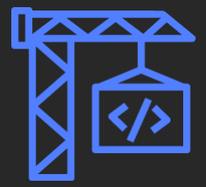


1. Automatically deploy new changes to staging environments for testing
2. Deploy to production safely without impacting customers
3. Deliver to customers faster

AWS Code Services

Source → Build → Test → Deploy

AWS CodeCommit



AWS CodeBuild

AWS CodeBuild
+ 3rd party tooling

AWS CodeDeploy

AWS CodePipeline

AWS Code Services

AWS CodeCommit

AWS CodeCommit

Secure, scalable, and managed Git source control

Git push

SSH or HTTPS

AWS CodeCommit

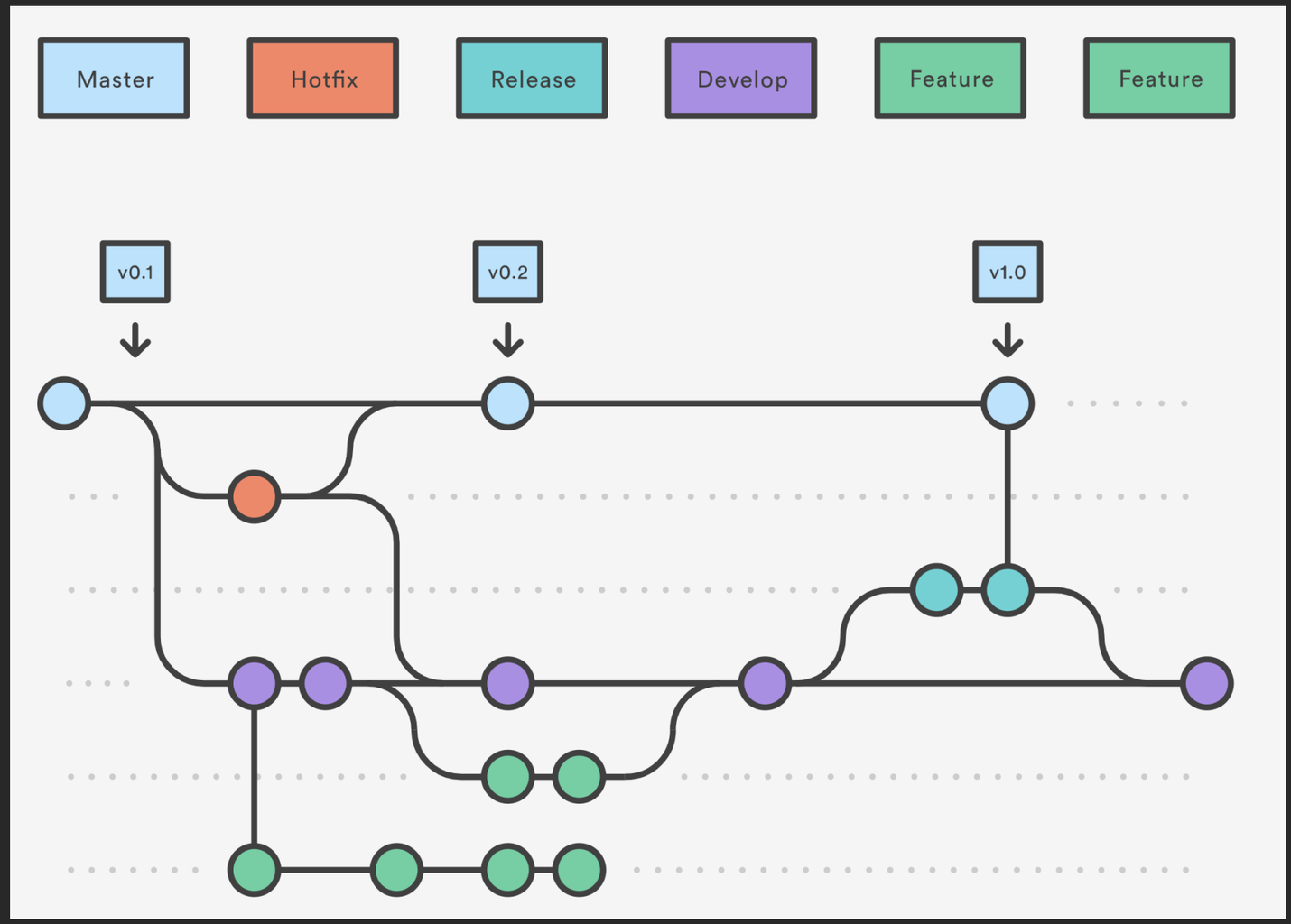
Git objects
in Amazon Simple
Storage Service
(Amazon S3)

Git index
in Amazon DynamoDB

Encryption keys
in AWS Key
Management Service
(KMS)

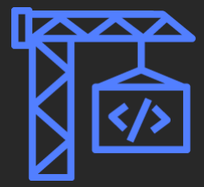
Branching strategy: GitFlow workflow

- Easier to understand and test
- Code review faster
- Backwards compatible
- GitFlow discipline



Credits: Atlassian GitFlow Workflow

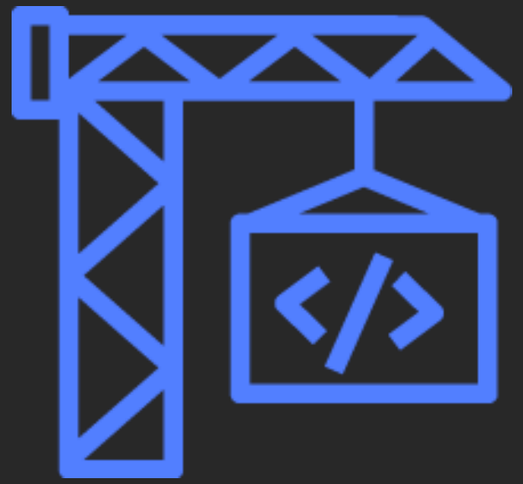
AWS Code Services



AWS CodeBuild

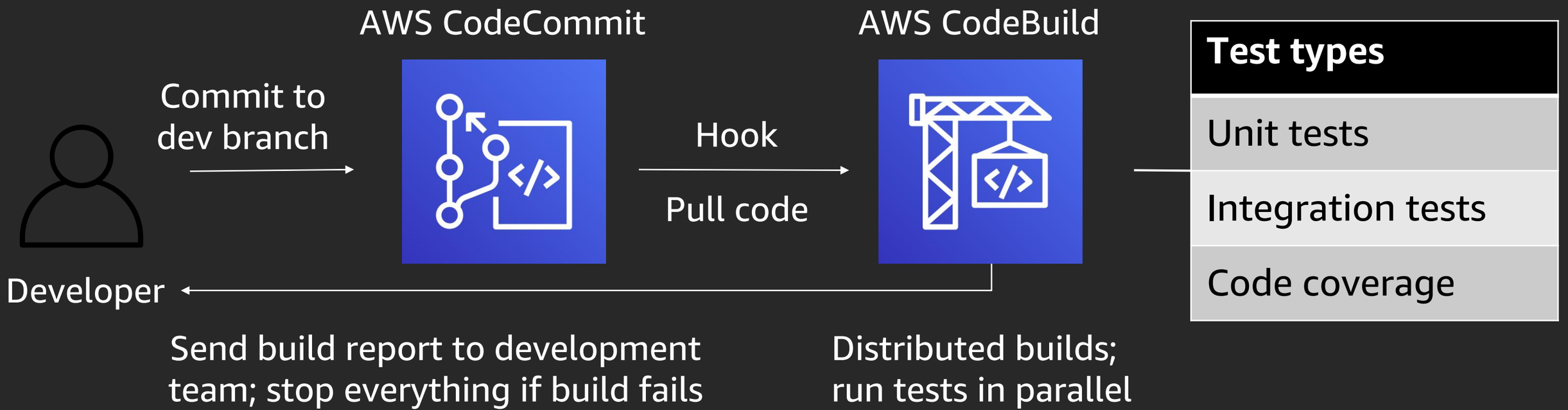
AWS CodeBuild
+ 3rd party tooling

AWS CodeBuild



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

Continuous integration workflow



AWS Code Services

AWS CodeDeploy

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

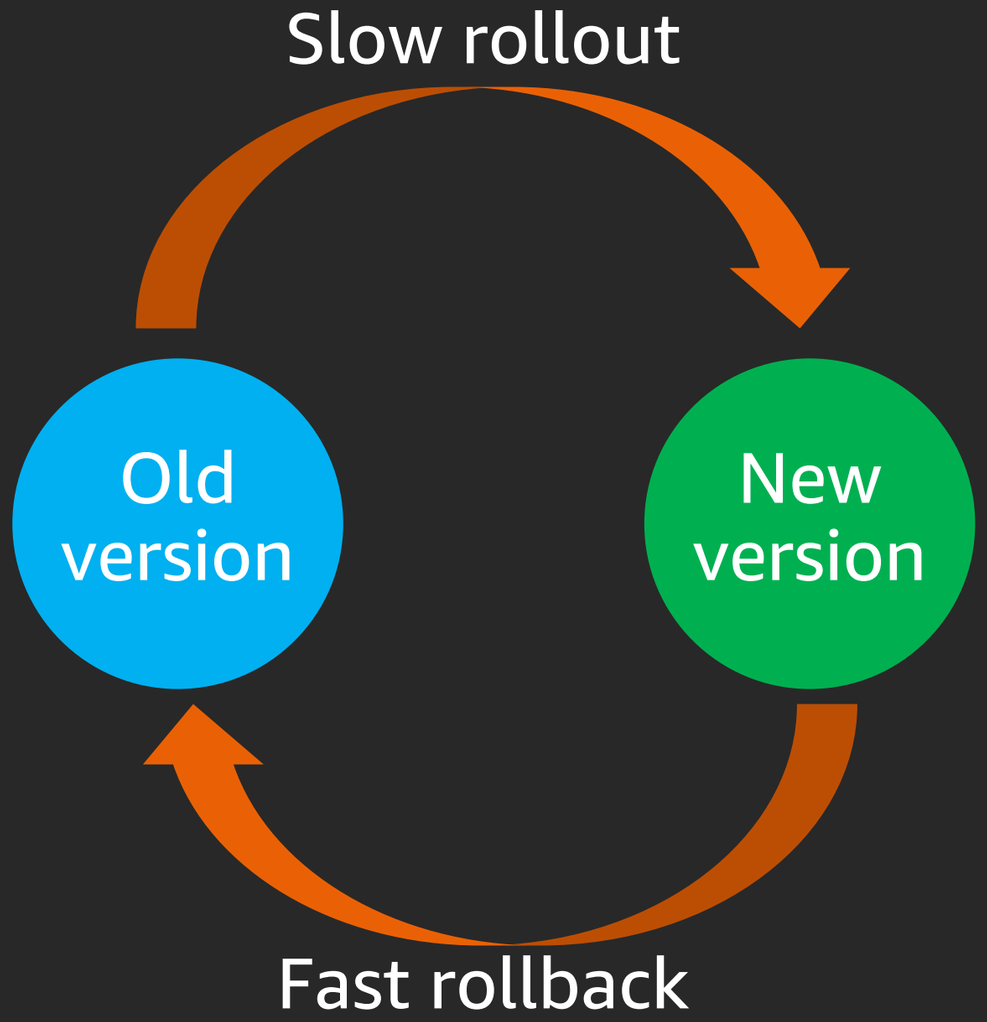
Choosing a deployment technique

Aim is to introduce the changes slowly

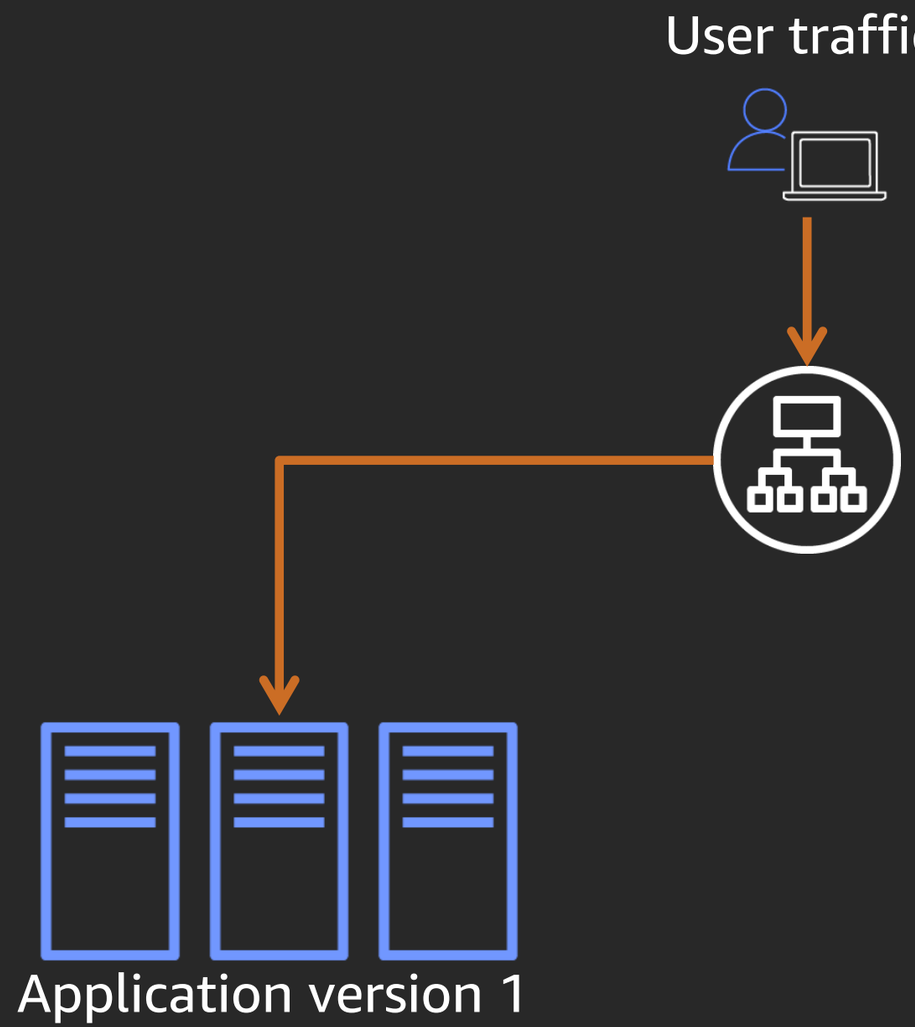
- Achieve zero downtime
- Deploy to only a few instances first
- Incrementally shift traffic to new version
- Bake and monitor performance before committing
- Achieve fast rollback on failures

Deployment strategies

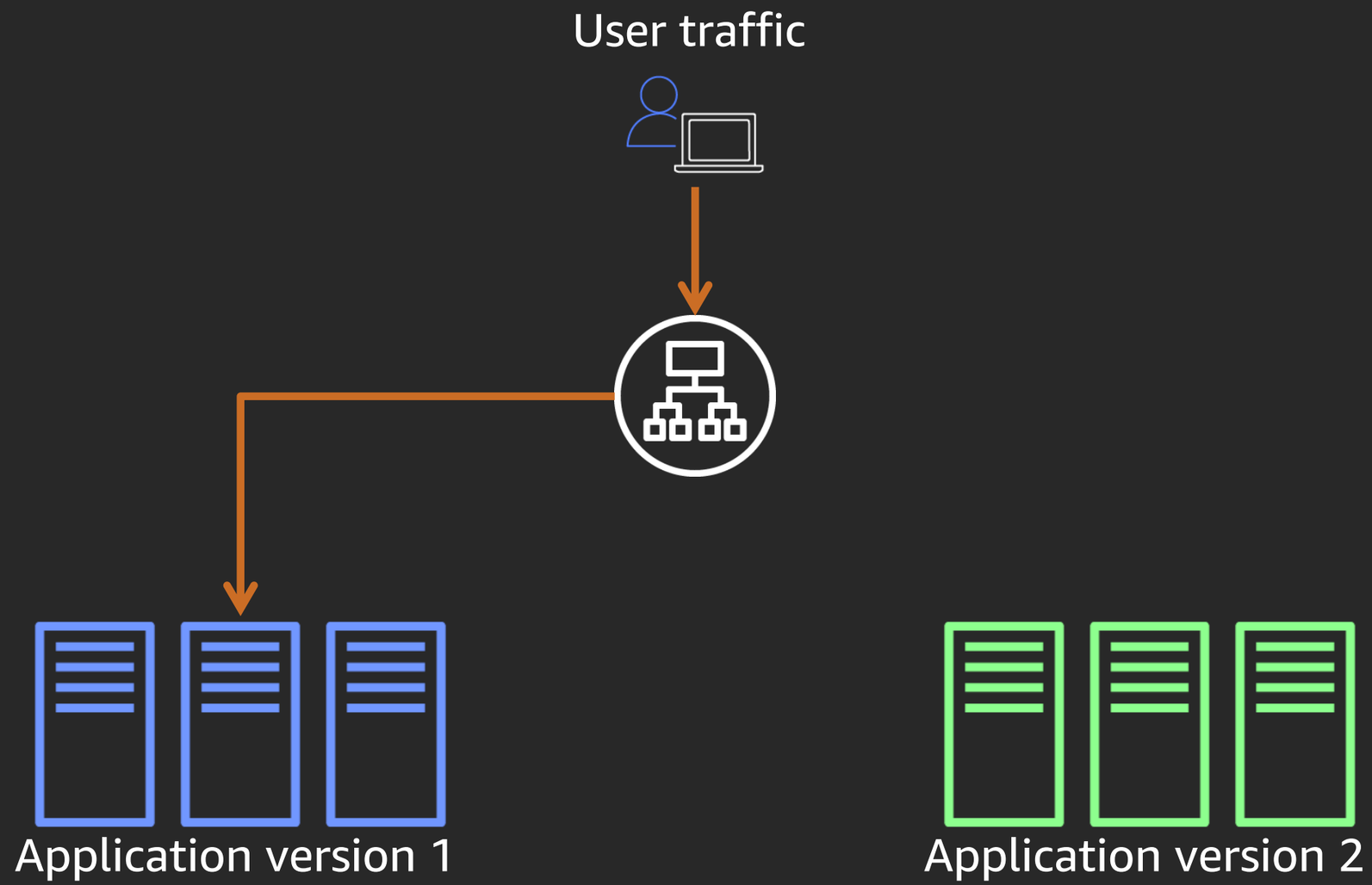
- Blue-Green
- Canary



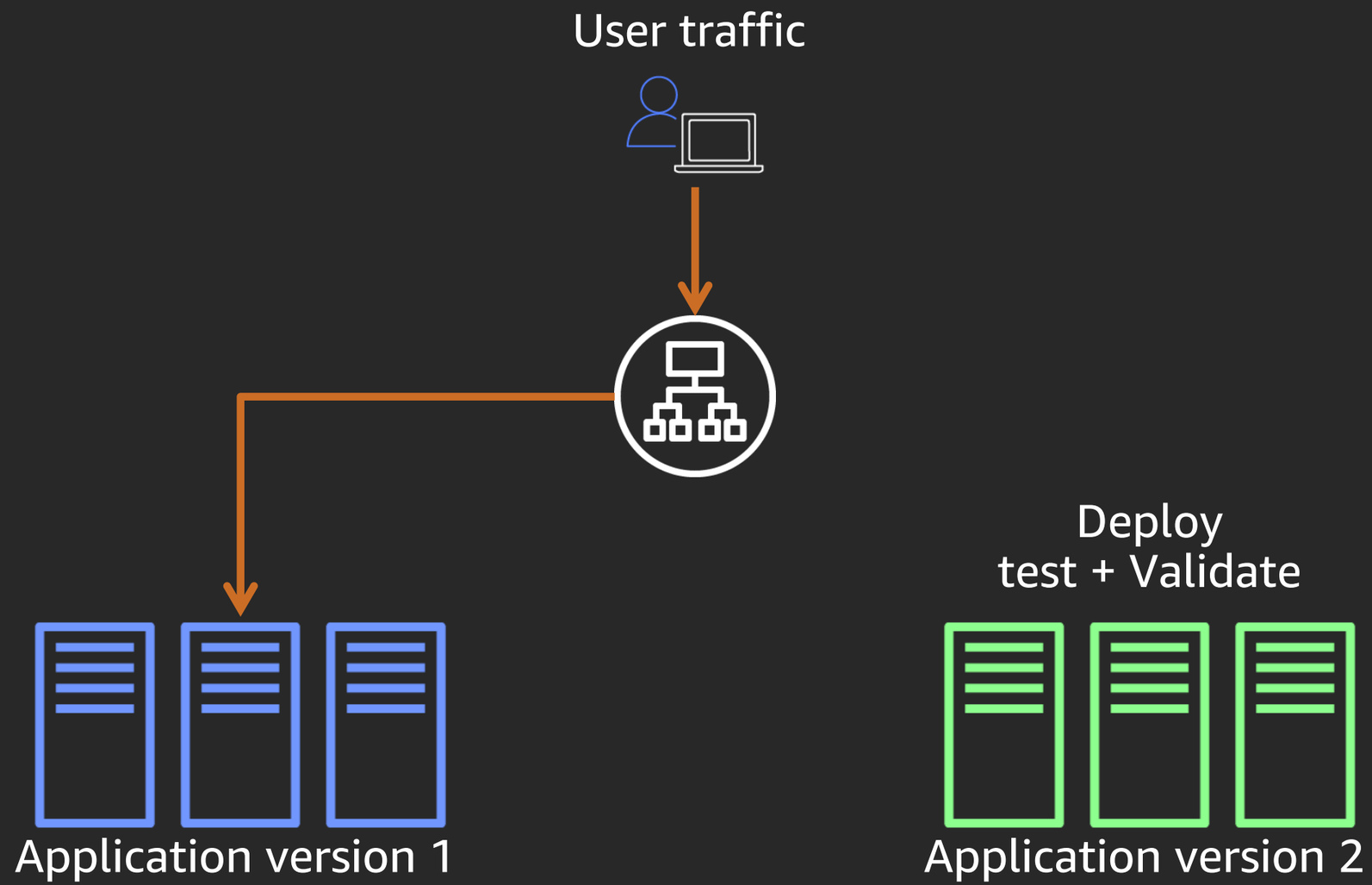
Blue-Green deployment



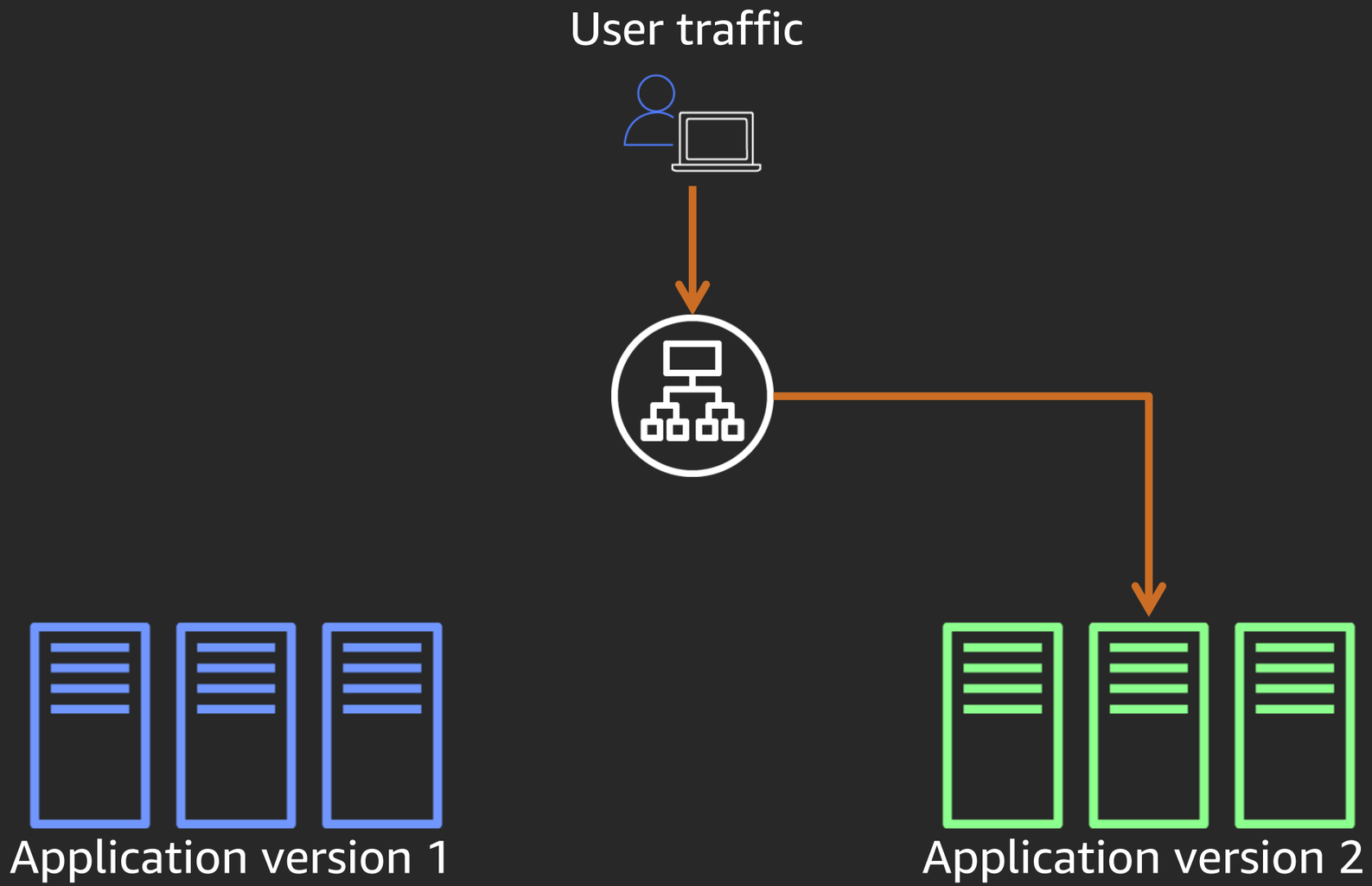
Blue-Green deployment



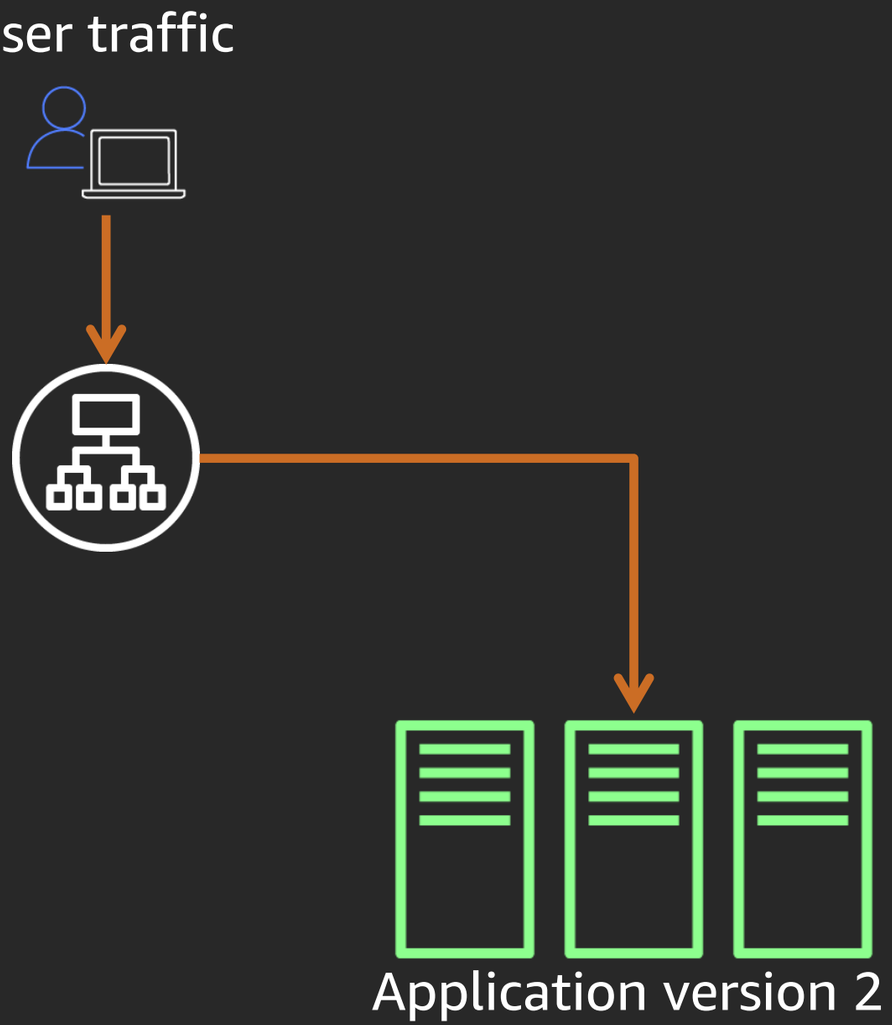
Blue-Green deployment



Blue-Green deployment



Blue-Green deployment



Blue-Green deployment

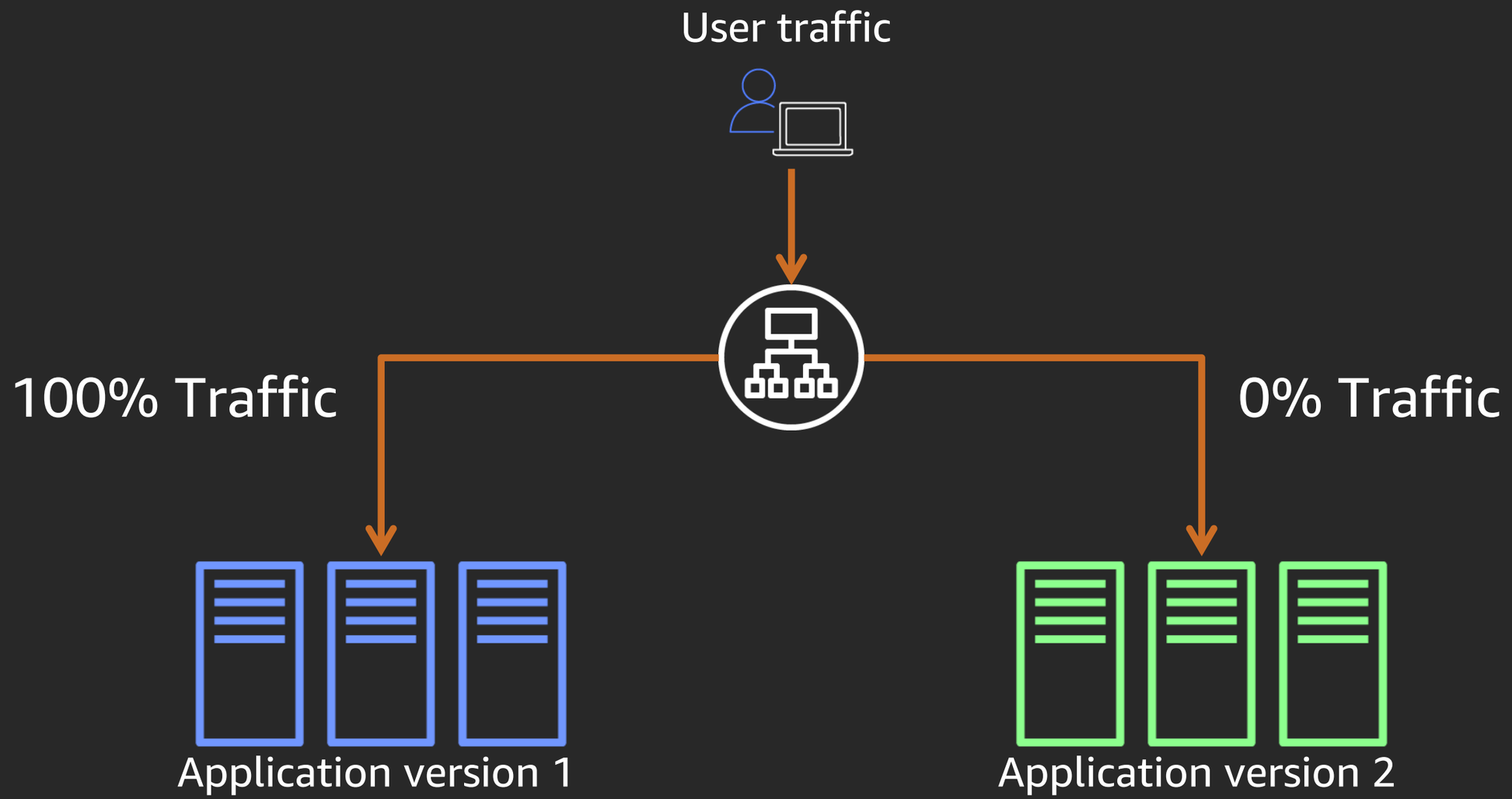
Advantages

- Zero downtime
- Original environment is not impacted
- Quick and safe rollback

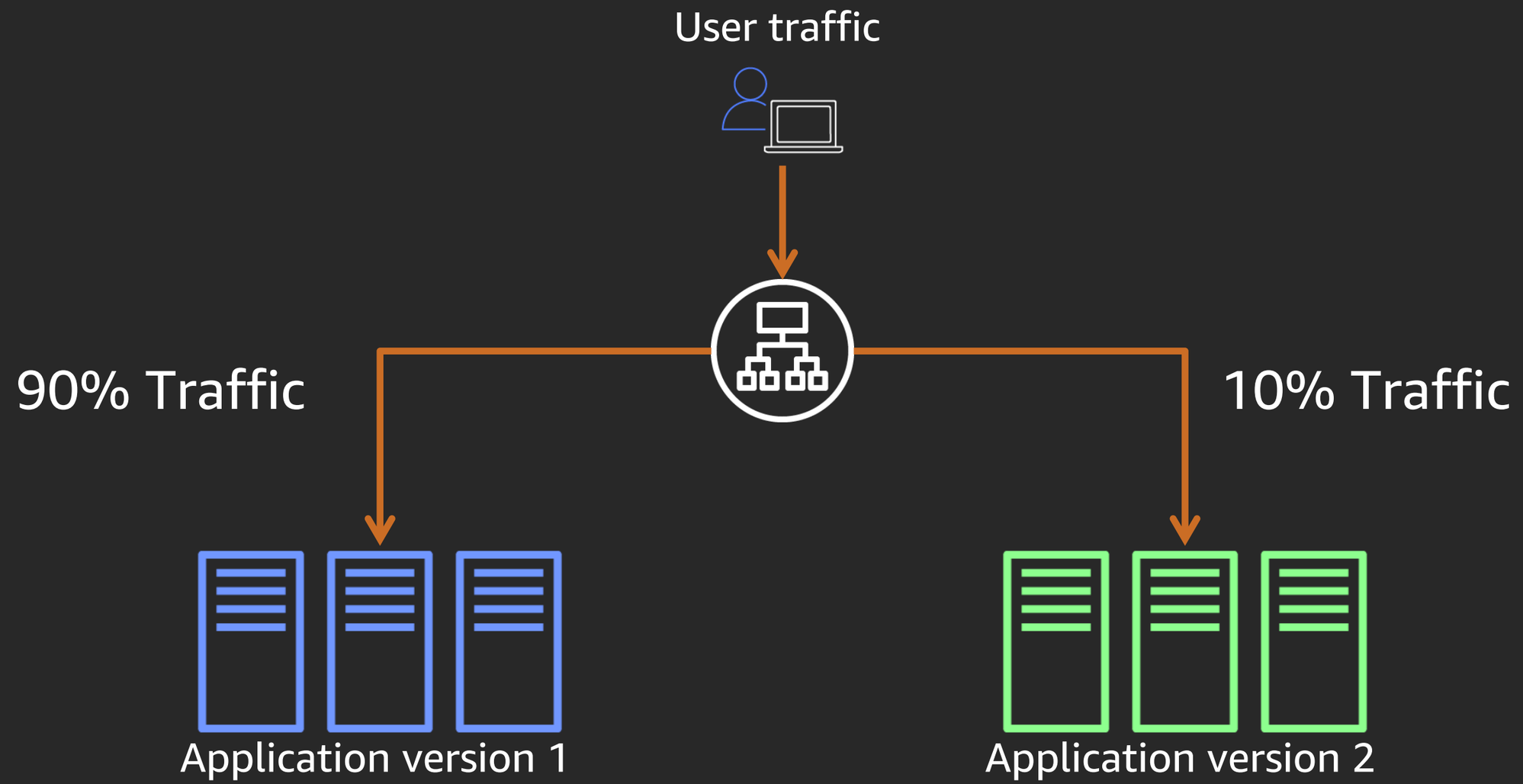
Disadvantages

- Added cost and time for creating new environment
- Operational overheads

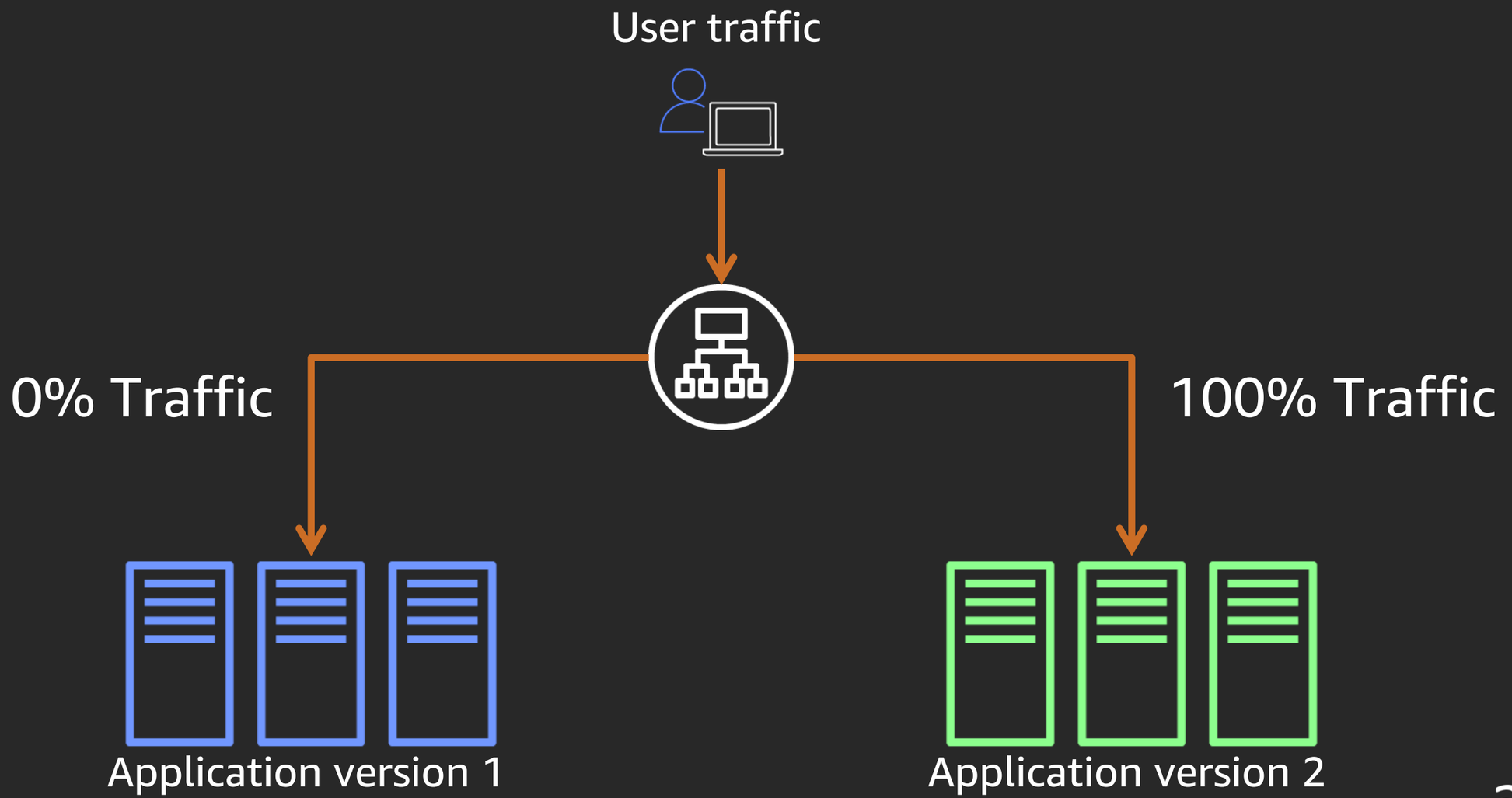
Canary deployment



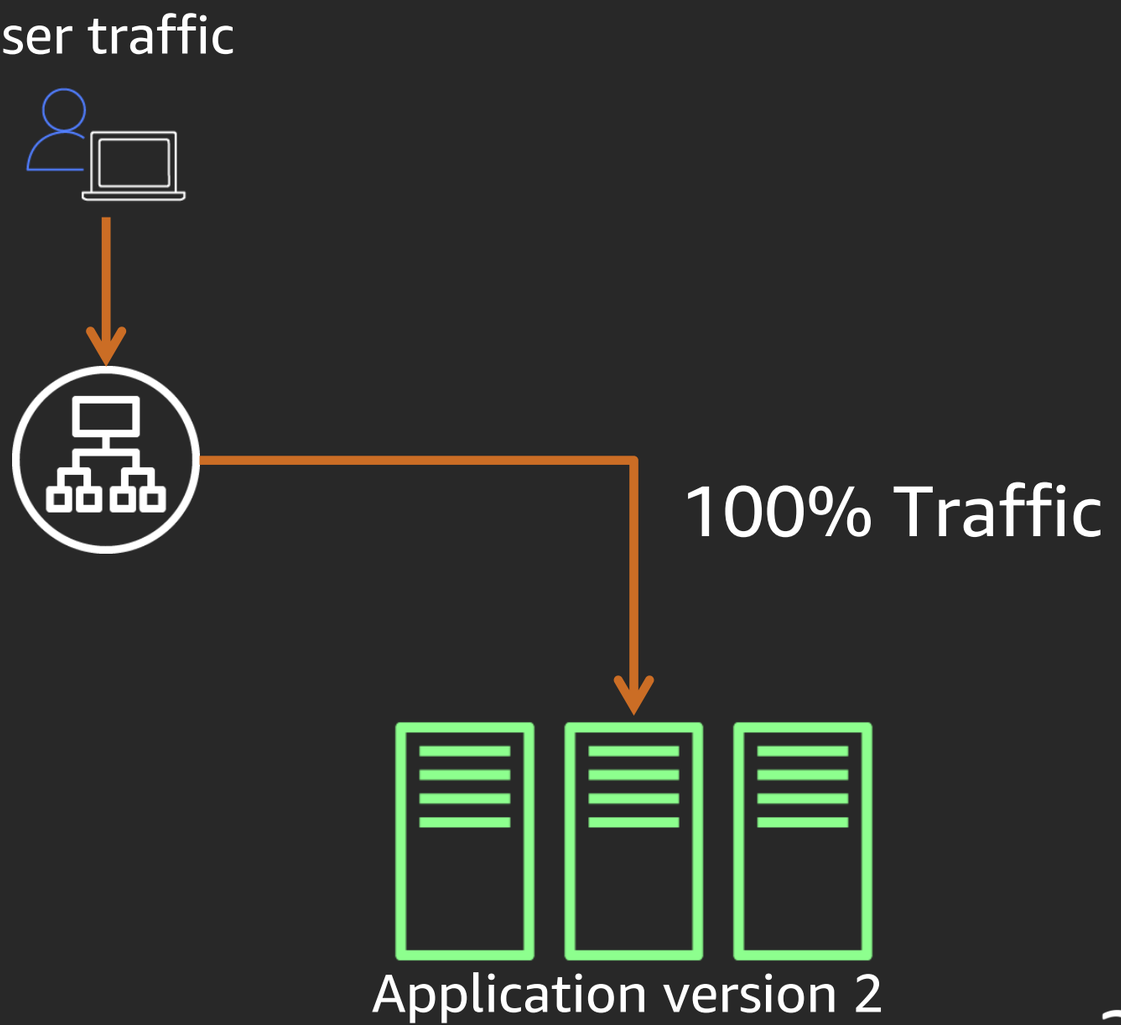
Canary deployment



Canary deployment



Canary deployment



Canary deployment

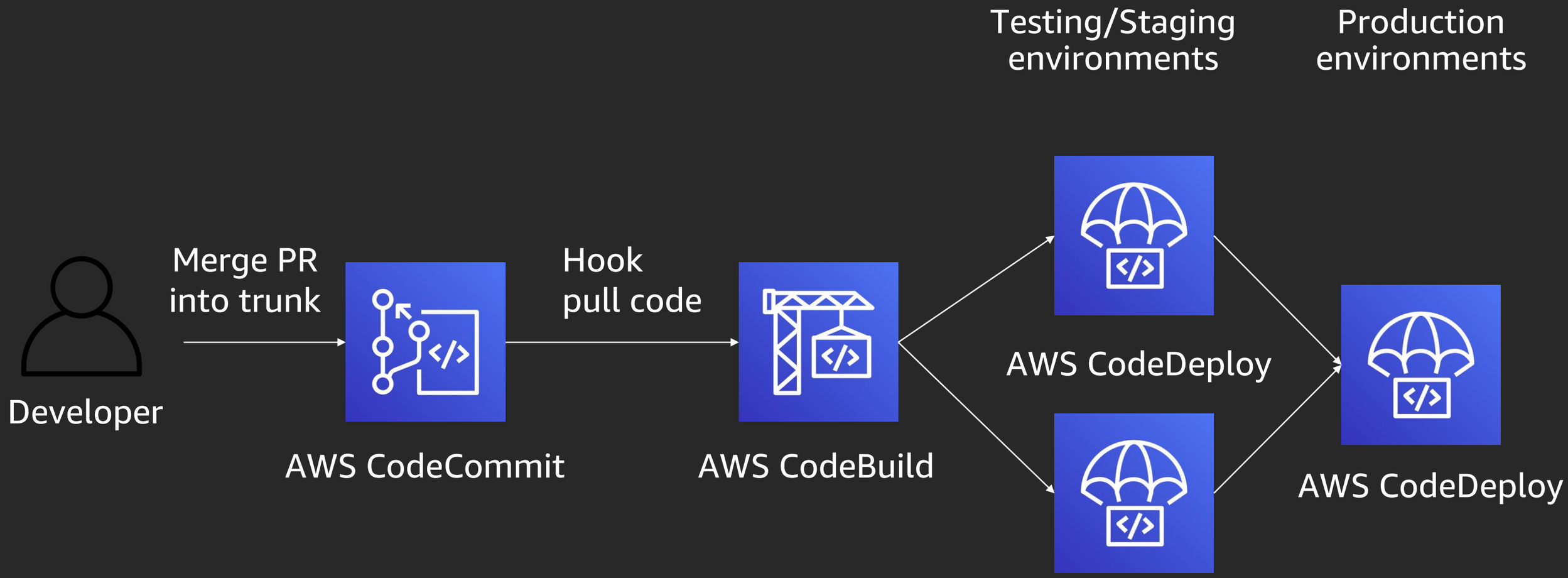
Advantages

- Test new version with small amount of production traffic before committing
- Quick and safe rollback
- Zero downtime

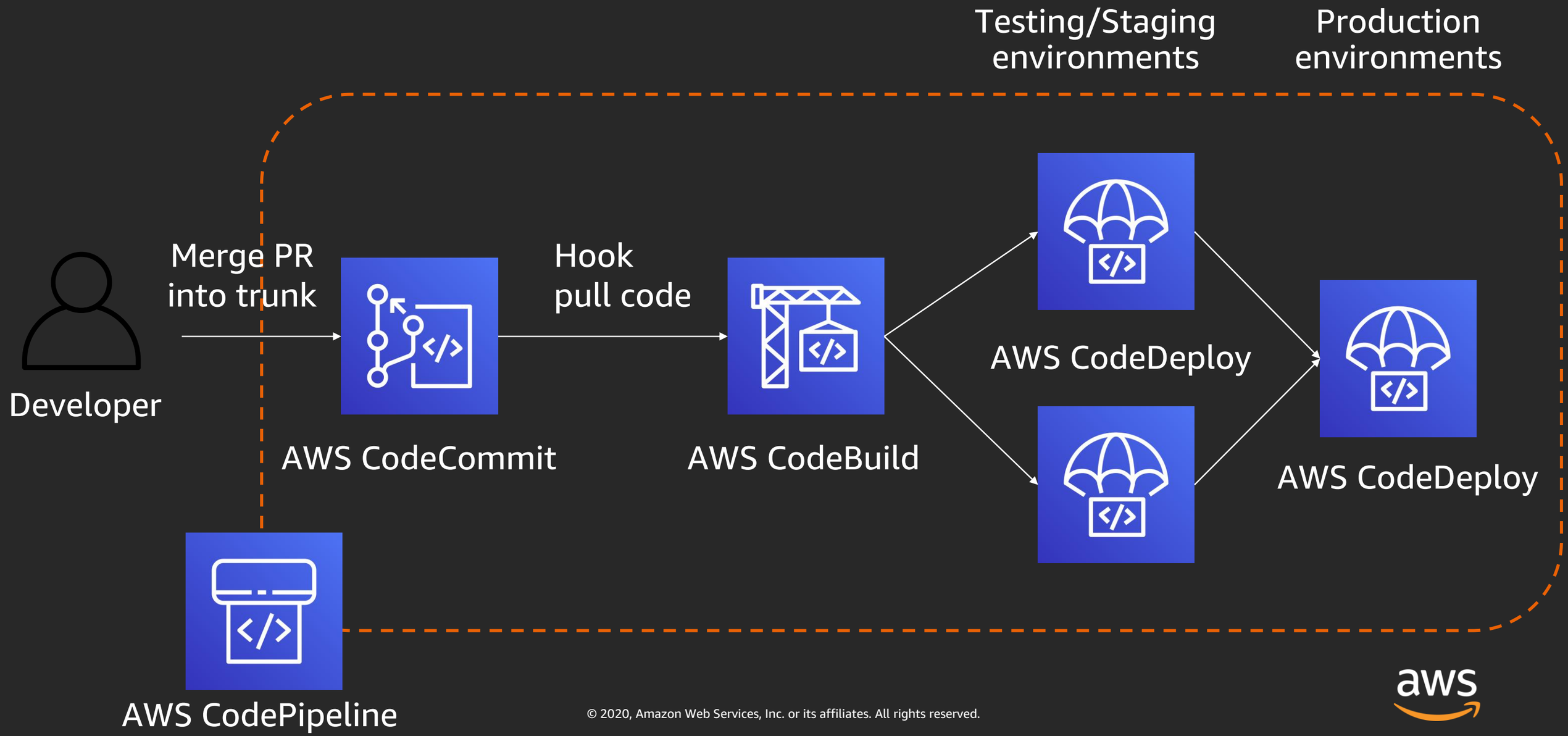
Disadvantages

- Old version and new version should be compatible
- Added cost and time for creating cloned environment

Continuous deployment workflow



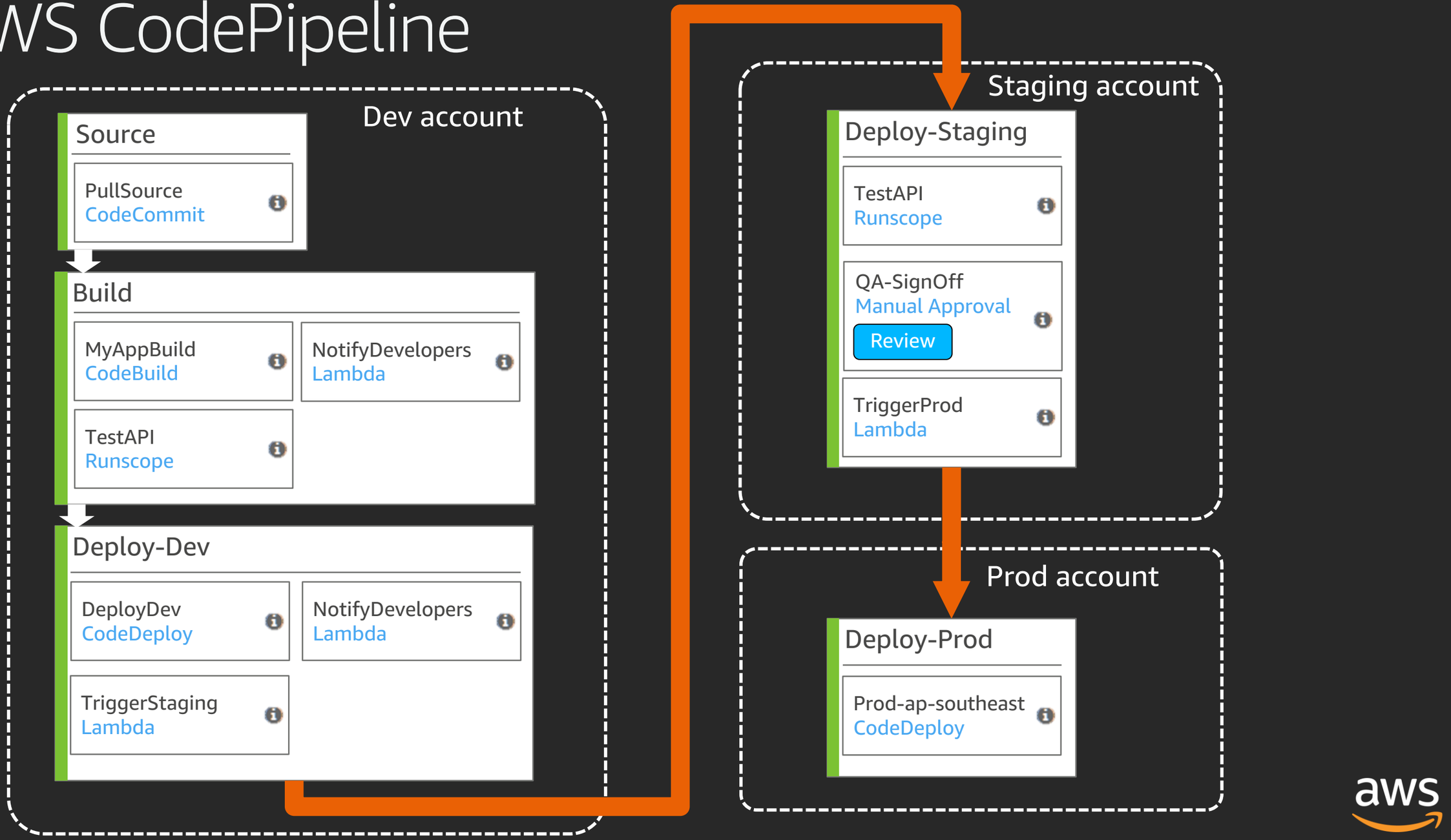
Continuous deployment workflow



AWS CodePipeline

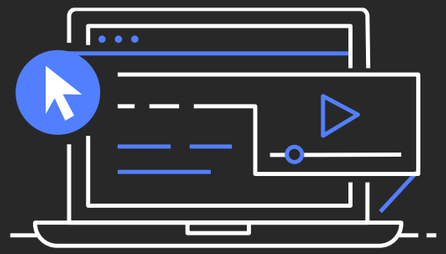
- Managed continuous delivery service
- Model and visualize release process
- Automated pipeline trigger on code change
- Integrates with third-party tools

AWS CodePipeline



Demo

AWS Digital Training



Flexibility to Learn Your Way

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

Featured Courses

- AWS Cloud Practitioner Essentials (Second Edition)
  Learn the fundamentals of the AWS Cloud and prepare for the AWS Certified Cloud Practitioner exam.
- Amazon DynamoDB for Serverless Architectures
  An introduction to Amazon DynamoDB and how it's leveraged in building a serverless architecture.
- AWS Security Fundamentals
  Learn fundamental cloud computing and AWS security concepts, including AWS access control and management, governance, logging, and encryption methods.
- Getting Started with Amazon Simple Storage Service (Amazon S3)
  The course provides you with the knowledge to determine when to use Amazon S3 by reviewing typical use cases and understanding how the service provides object storage for your applications.

aws.amazon.com/training

Thank you for attending AWS Builders 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

Builders Online Series

Thank you