
AWS
re:Invent

DOP307

Orchestrating complex deployments using AWS Developer Tools

Leo Zhadanovsky

Principal Solutions Architect

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Overview of AWS CodePipeline and AWS Step Functions

Types of actions for complex deployments

Orchestrating complex deployment scenarios

Demo

Q&A

Next steps to learn more



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Overview of AWS CodePipeline and AWS Step Functions



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Users tell us they want . . .

. . . it to be easy for their development teams to move to the cloud and deliver with AWS

. . . tools that help them adopt modern application development techniques so they can get the most out of the cloud

The AWS Developer Tools

The 2018 Accenture Developer Ecosystem Survey named AWS the **“most developer friendly”** and **“most forward looking”**

Comprehensive set of enterprise-grade tools	AWS has a robust set of tools designed to support a wide array of use cases at scale
Built for modern application development	Designed to support modern application development techniques (such as CI / CD & infrastructure as code) and technologies (such as serverless & containers)

Comprehensive set of enterprise-grade tools

CI / CD tools



Infrastructure as code



IDE



Monitoring and tracing



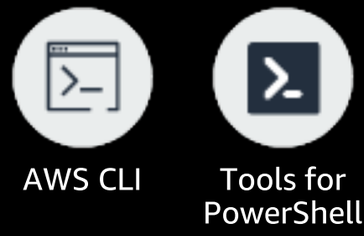
Web apps



IDE and DevOps toolkits



CLI and scripting tools



Languages



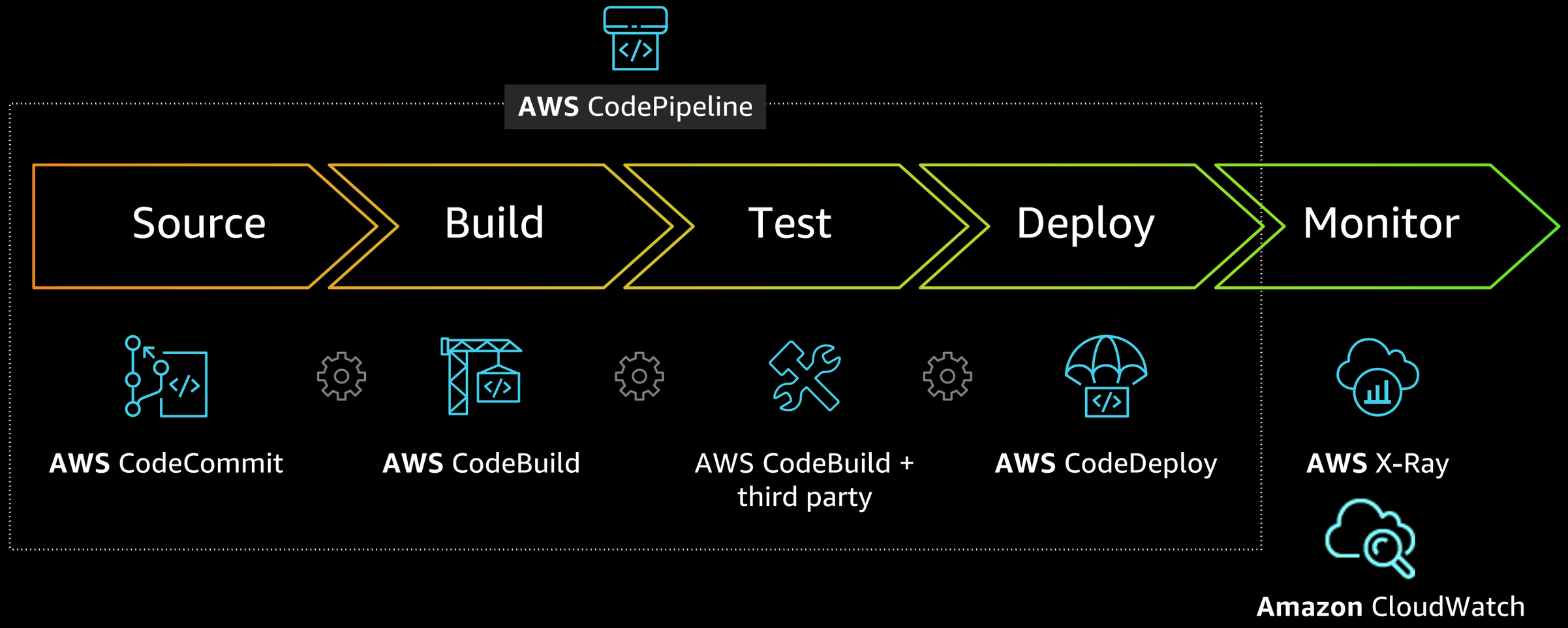
Mobile



SDKs



The AWS Developer Tools for CI / CD



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

CodePipeline: Supported sources

Automatically kick off release and pull latest source code

Pick branch

CodeCommit

GitHub

GitHub Enterprise

Bitbucket

Pick object, folder or container image

Amazon Simple Storage Service (Amazon S3)

Amazon Elastic Container Registry (Amazon ECR)

CodePipeline: Supported deployment targets

Automatically kick off release and pull latest source code

Amazon EC2

CodeDeploy

AWS Elastic Beanstalk

AWS OpsWorks stacks

Containers

CodeDeploy

Amazon Elastic Container Service (Amazon ECS)

Amazon ECS (blue / green deployment)

AWS Fargate

Serverless

CodeDeploy

AWS CloudFormation / AWS SAM

AWS Lambda

AWS Step Functions

AWS Step Functions

FULLY-MANAGED STATE MACHINES ON AWS

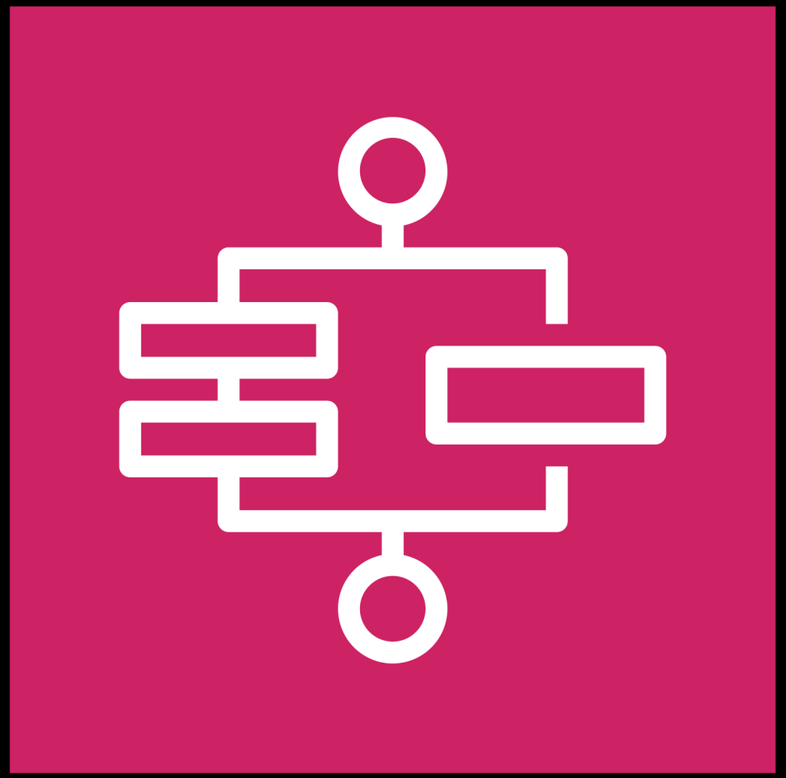
Resilient workflow automation

Built-in error handling

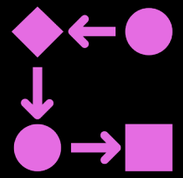
Powerful AWS service integration

First-class support for integrating with your own services

Auditable execution history and visual monitoring



How AWS Step Functions work



The workflows you build with Step Functions are called state machines, and each step of your workflow is called a state



When you execute your state machine, each move from one state to the next is called a state transition



You can reuse components, easily edit the sequence of steps, or swap out the code called by task states as your needs change

Amazon States Language

https://states-language.net/spec.html

```
{
    "Comment": "A simple minimal example",
    "StartAt": "Hello World",
    "States": {
      "Hello World": {
        "Type": "Task",
        "Resource": "arn:aws:lambda...HelloWorld",
        "End": true
      },
      [. . .]
  }
}
```

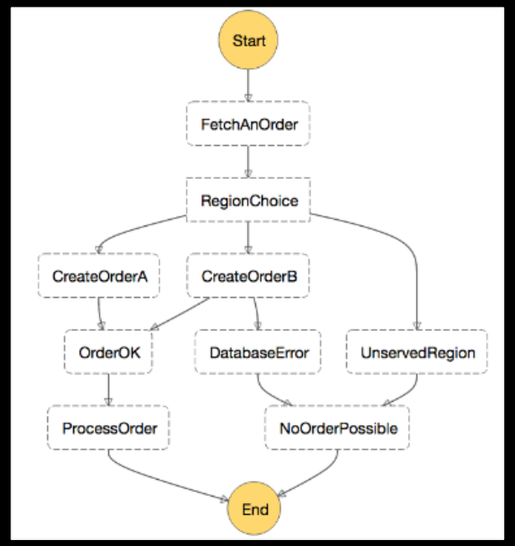


Working with AWS Step Functions

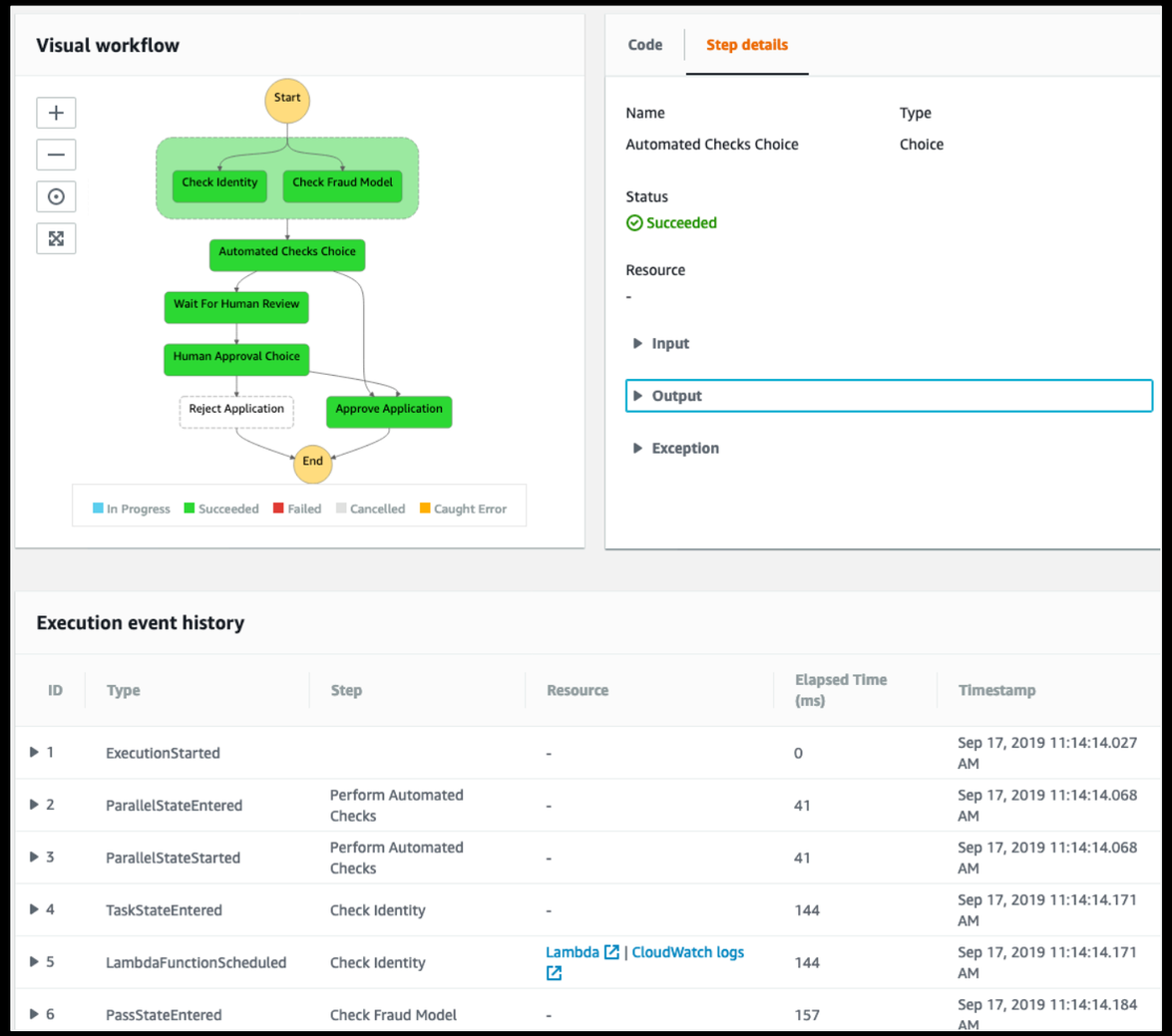
Define in JSON

```
{
    "Comment": "Manage opening an account",
    "StartAt": "Perform Automated Checks",
    "States": {
        "Perform Automated Checks": {
            "Type": "Parallel",
            "Branches": [{
                    "StartAt": "Check Identity",
                    "States": {
                        "Check Identity": {
                            "Type": "Task",
                            "Parameters": {
```

Visualize in the console

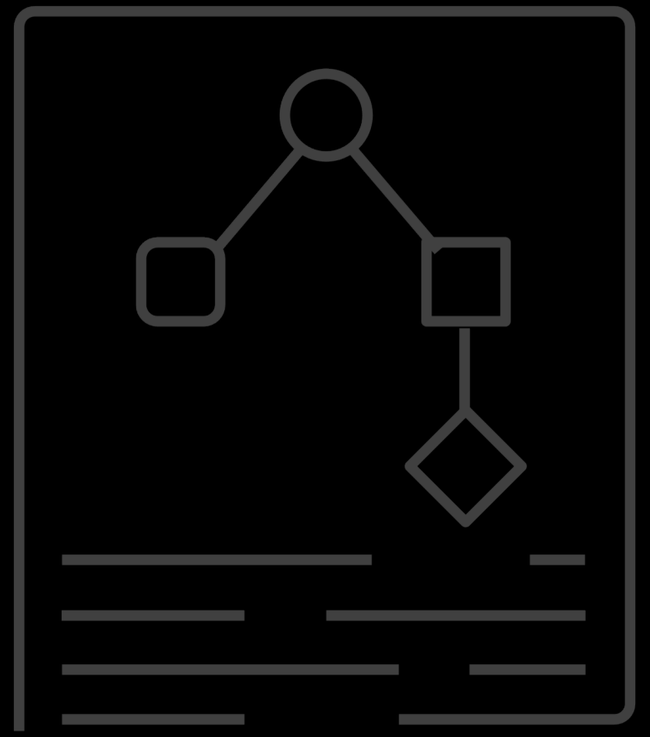


Monitor executions



State types

Task	Execute work
Choice	Add branching logic
Wait	Add a timed delay
Parallel	Execute branches in parallel
Map	Process each of an input array's items with a state machine
Succeed	Signal a successful execution and stop
Fail	Signal a failed execution and stop
Pass	Pass input to output



Types of deployments



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Types of deployments

Simple

- No complex state management
- Whole pipeline can be done with existing integrations and Lambda custom actions
- Data across stages and actions entirely done through input and output artifacts

Complex

- Fine-grained customization required
- State management
- Marshalling of data across stages and actions

Types of actions for complex deployments

Lambda custom action

- Triggers Lambda function
- Great for simple actions
  - Trigger an API
  - Write value to an Amazon DynamoDB table
  - Automate manual approvals
- Maximum execution time: 15 minutes

Step Functions action

- Triggers Step Functions state machine execution
- Great for complex actions, requiring orchestration, decision-making, and complex logic
  - Trigger a distributed load test
  - Run an Amazon ECS task until some arbitrary condition is satisfied
  - Read and write values to DynamoDB tables as atomic transactions
- Maximum execution time: 365 days

Orchestrating complex deployment scenarios



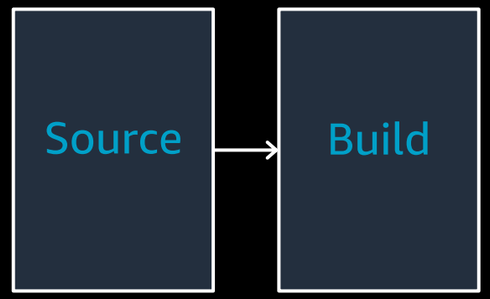
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

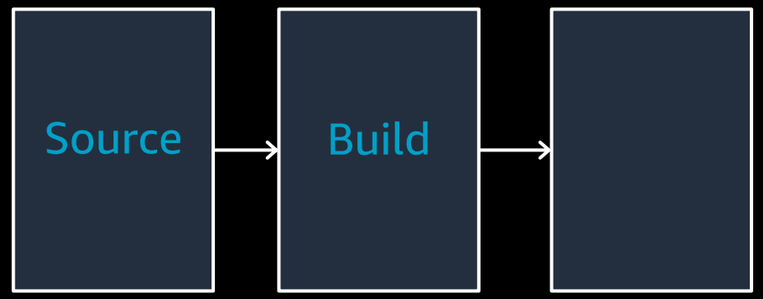
Amazon continuous delivery: Deep dive



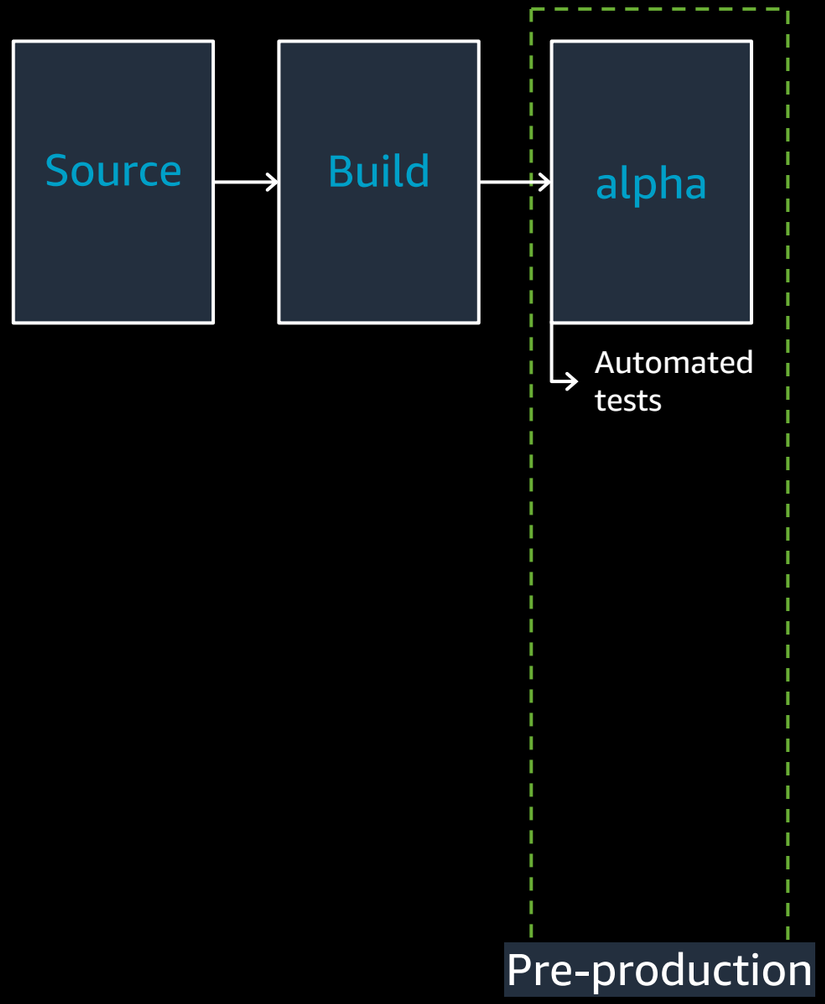
Amazon continuous delivery: Deep dive



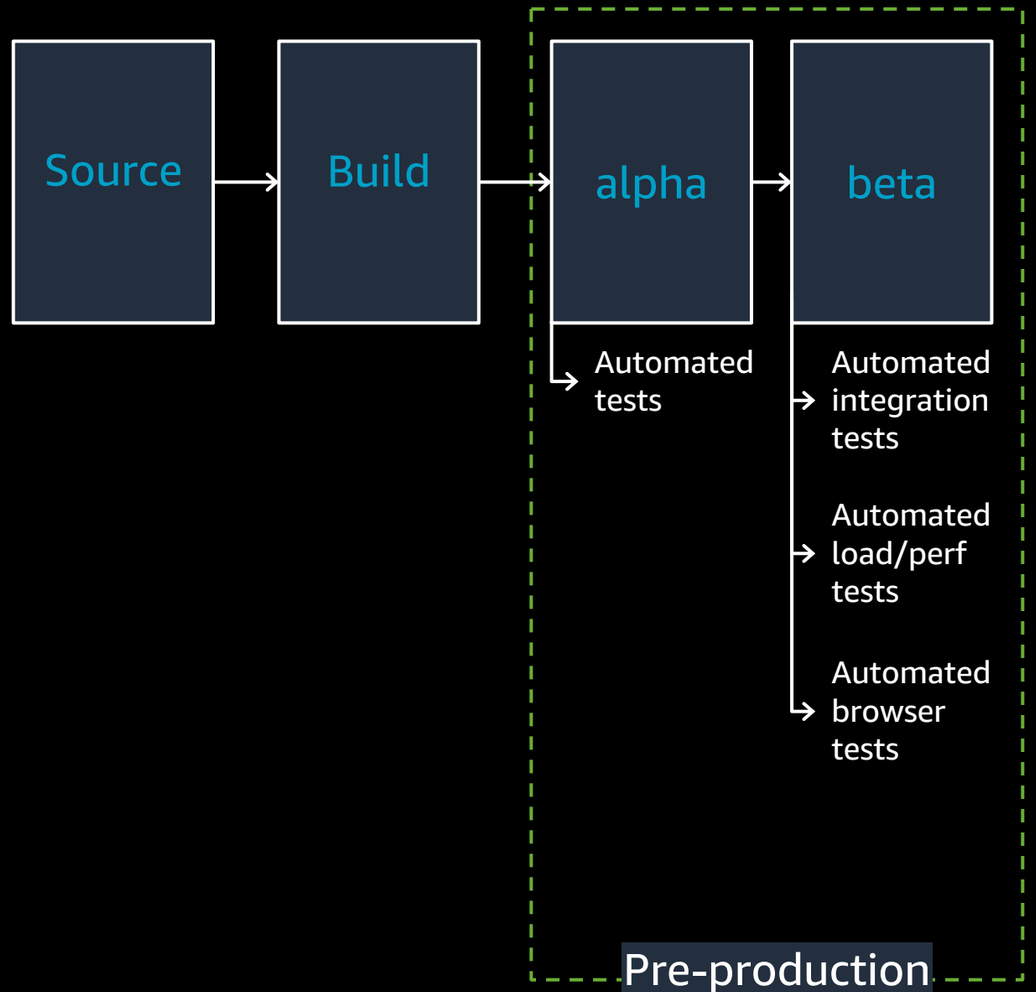
Amazon continuous delivery: Deep dive



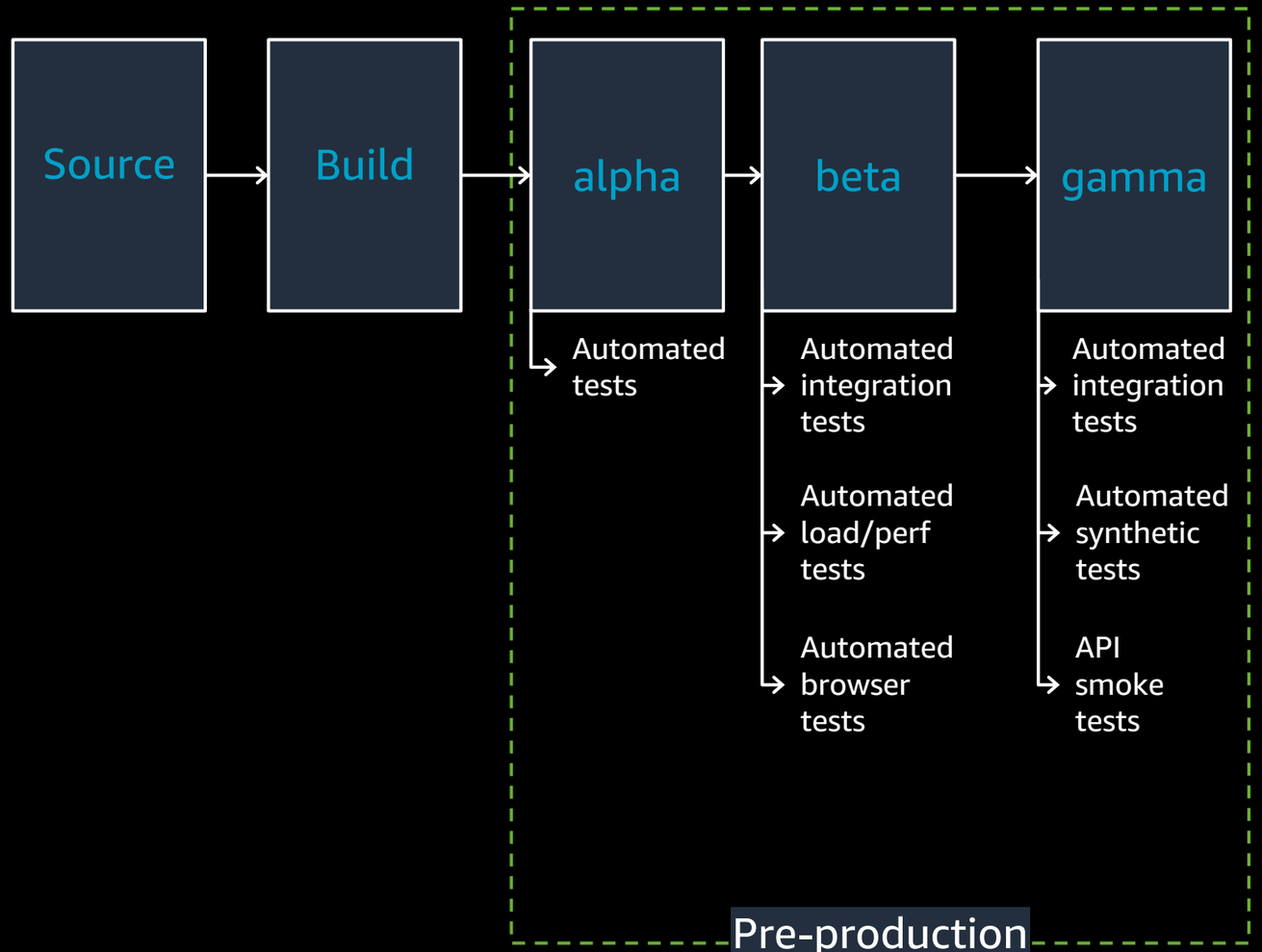
Amazon continuous delivery: Deep dive



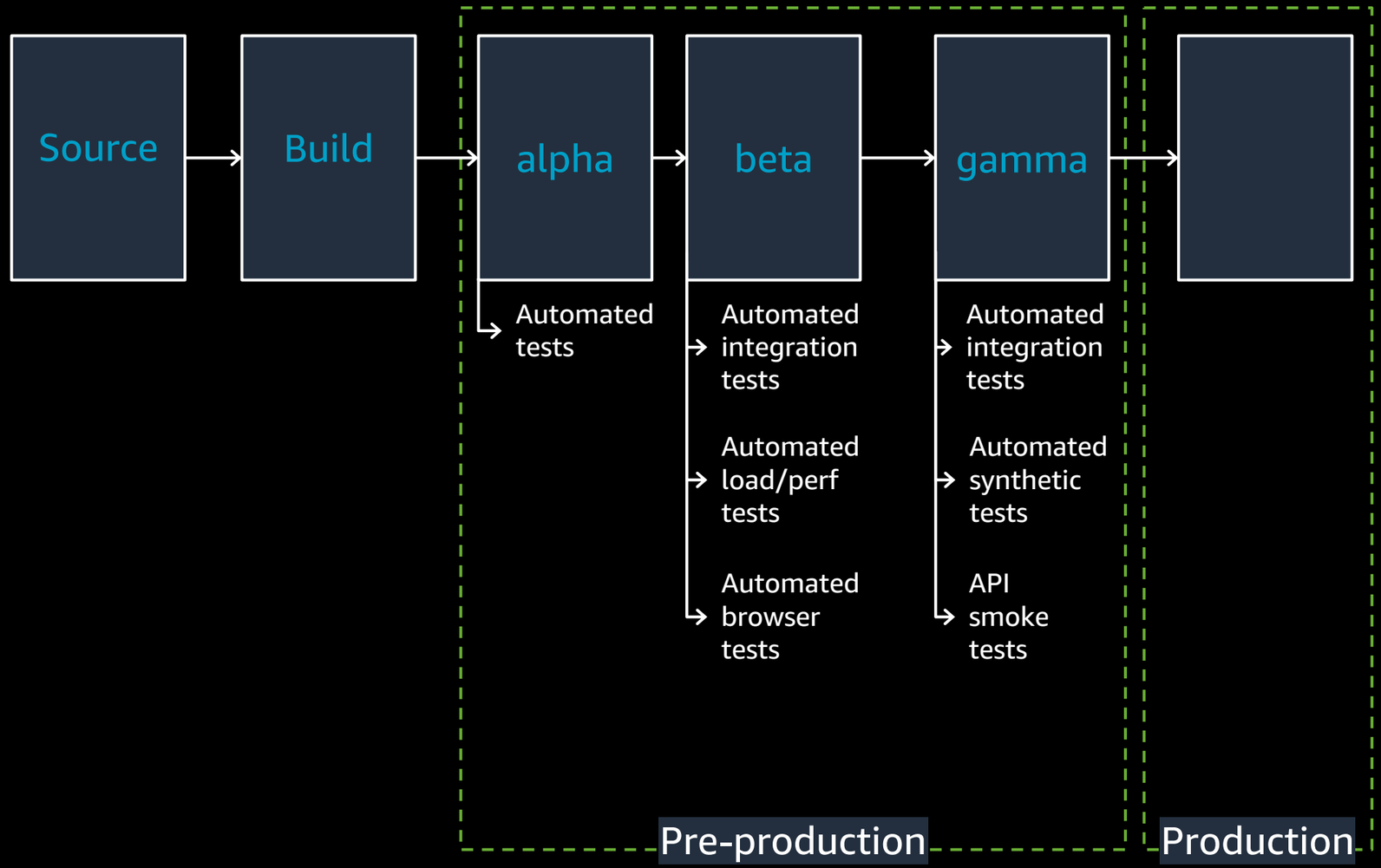
Amazon continuous delivery: Deep dive



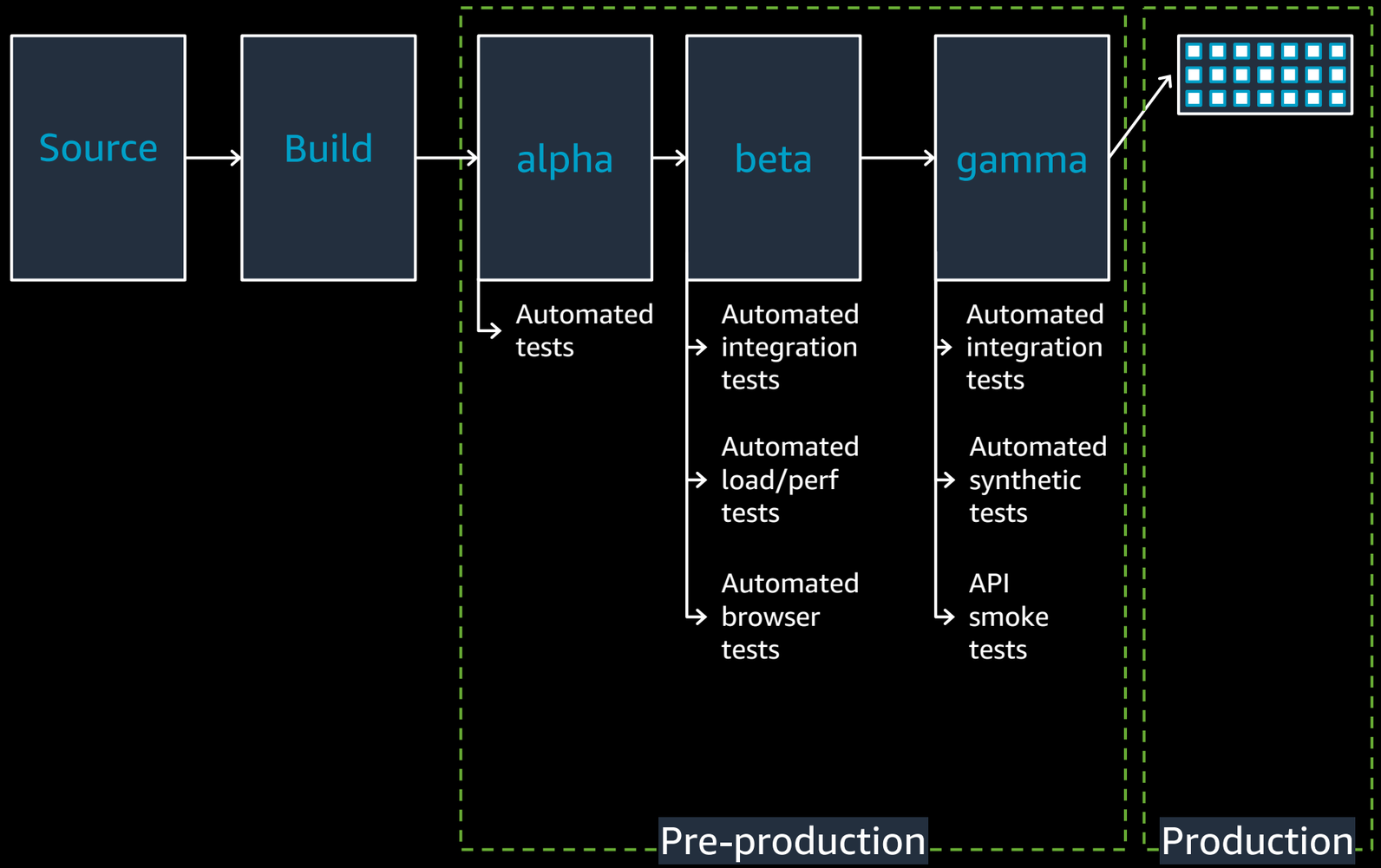
Amazon continuous delivery: Deep dive



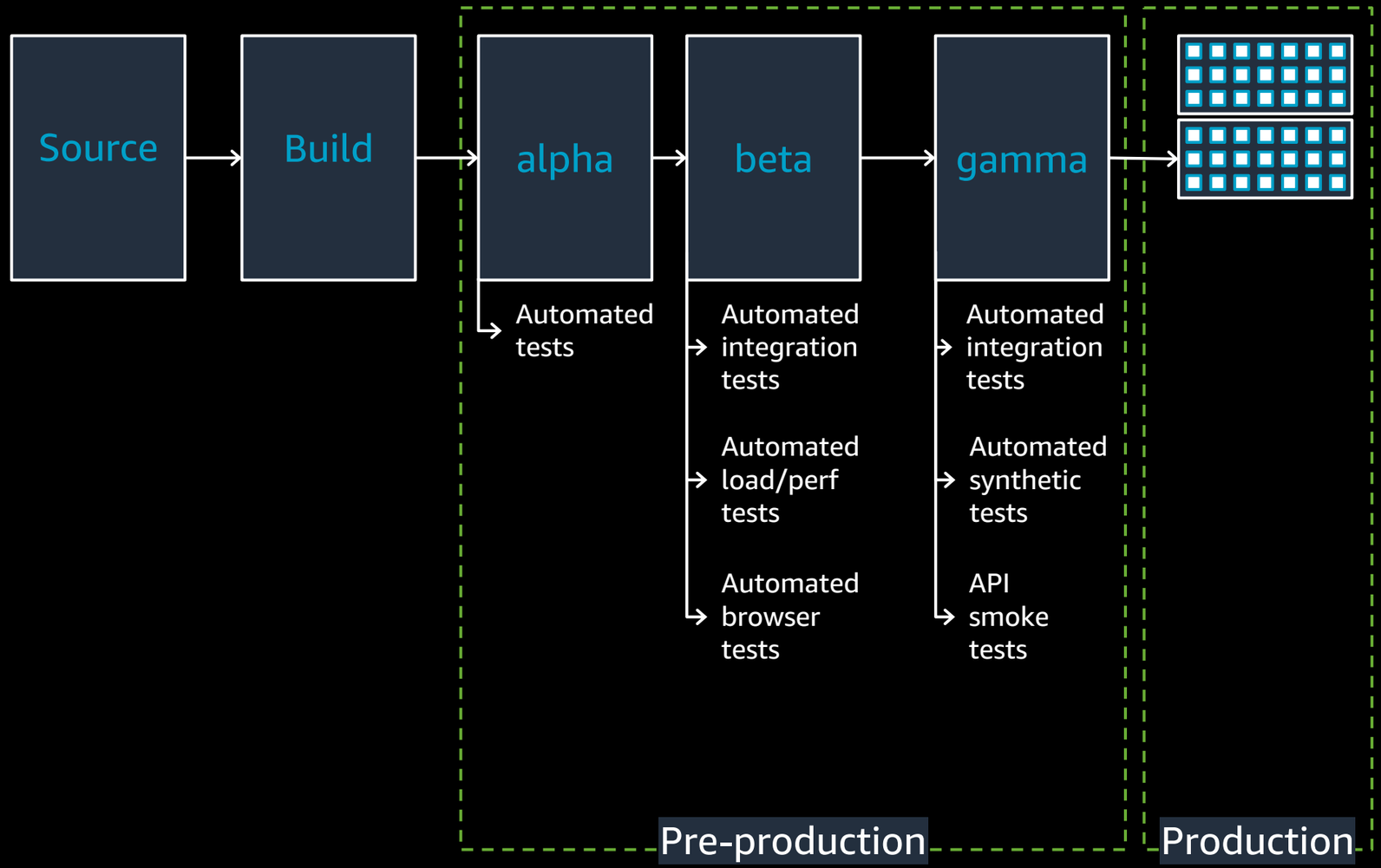
Amazon continuous delivery: Deep dive



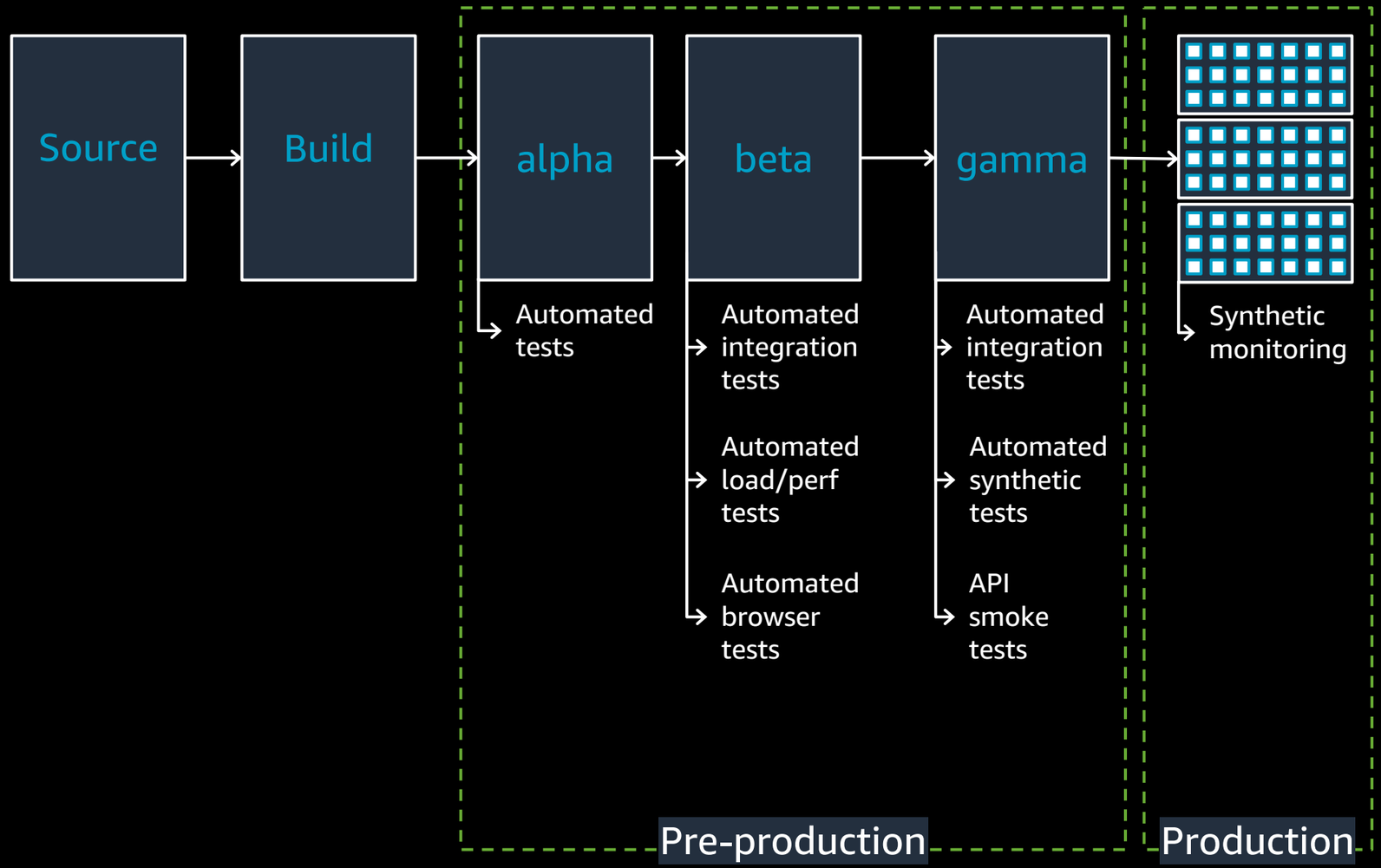
Amazon continuous delivery: Deep dive



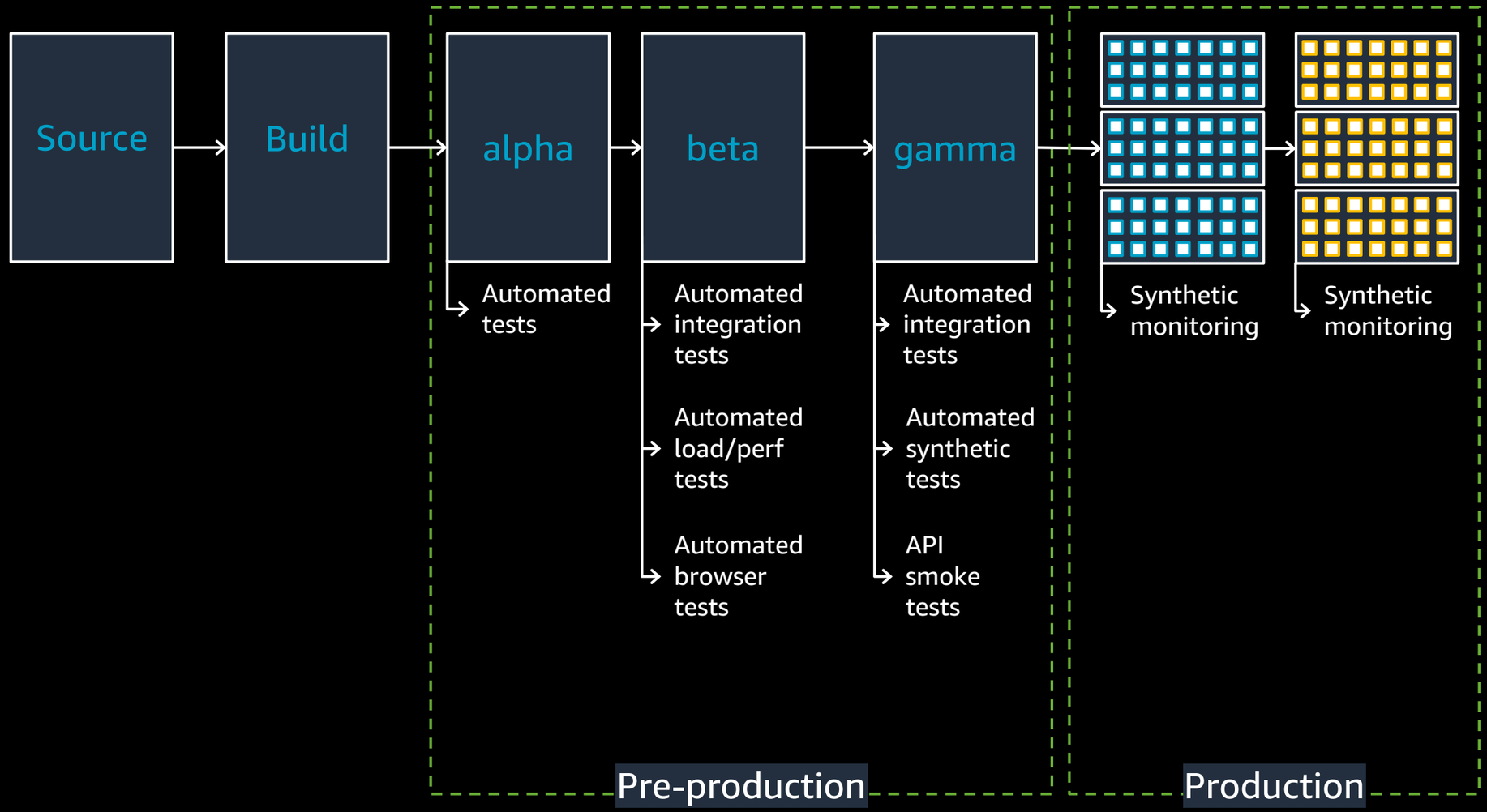
Amazon continuous delivery: Deep dive



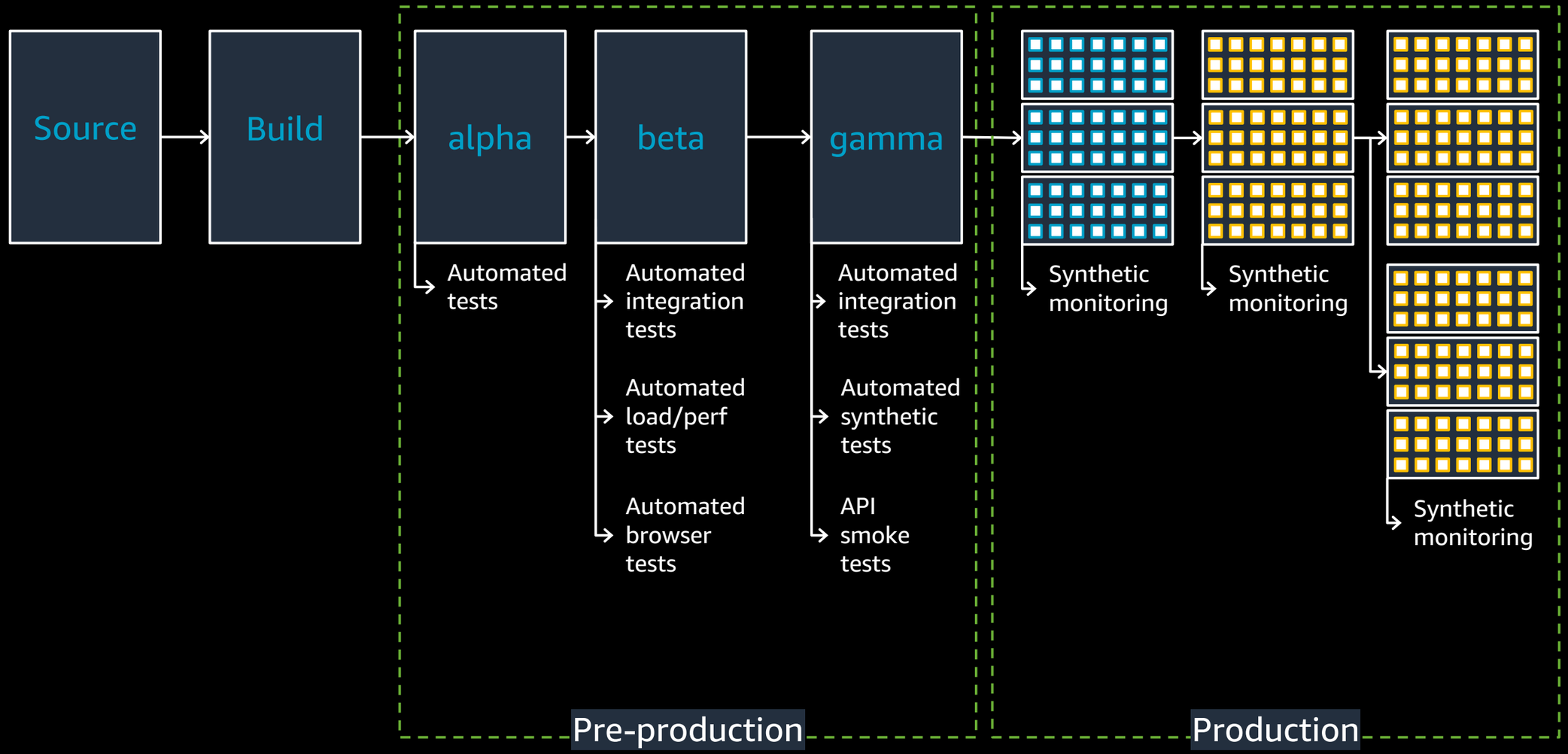
Amazon continuous delivery: Deep dive



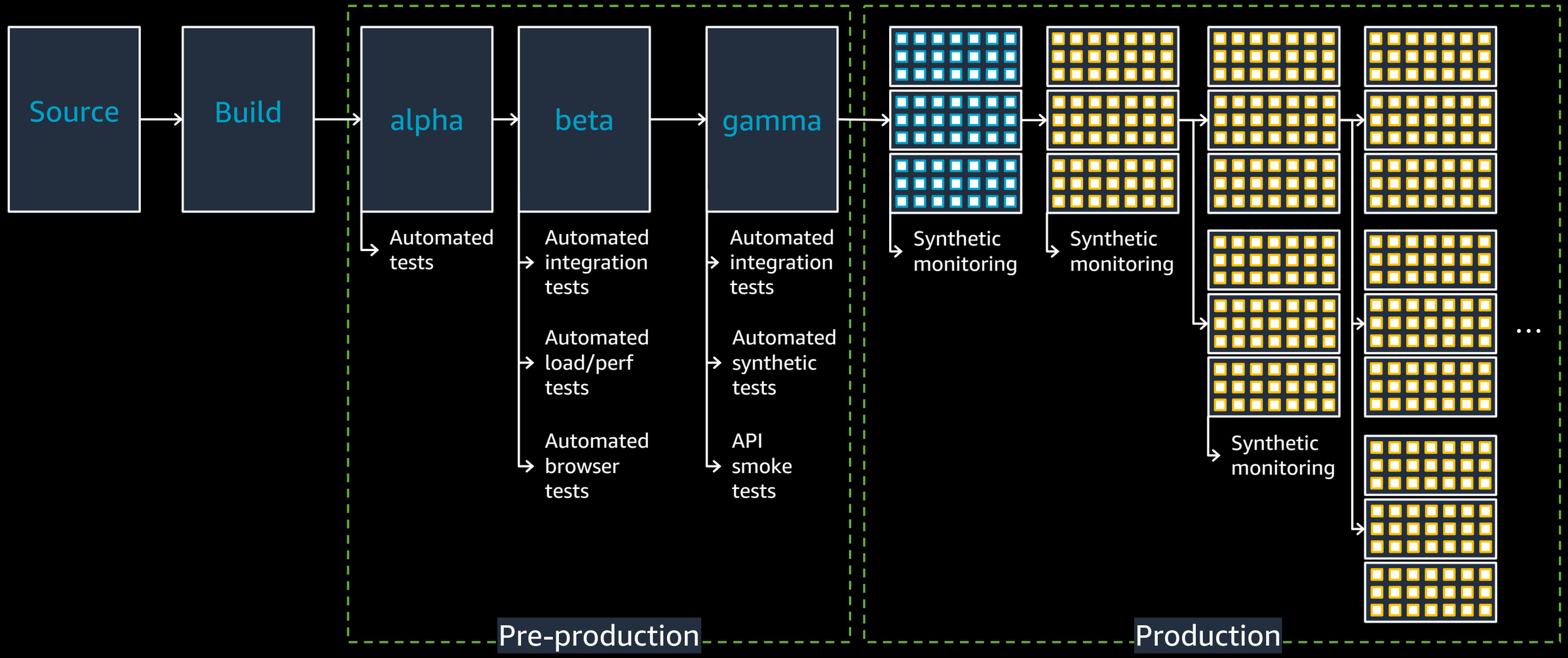
Amazon continuous delivery: Deep dive



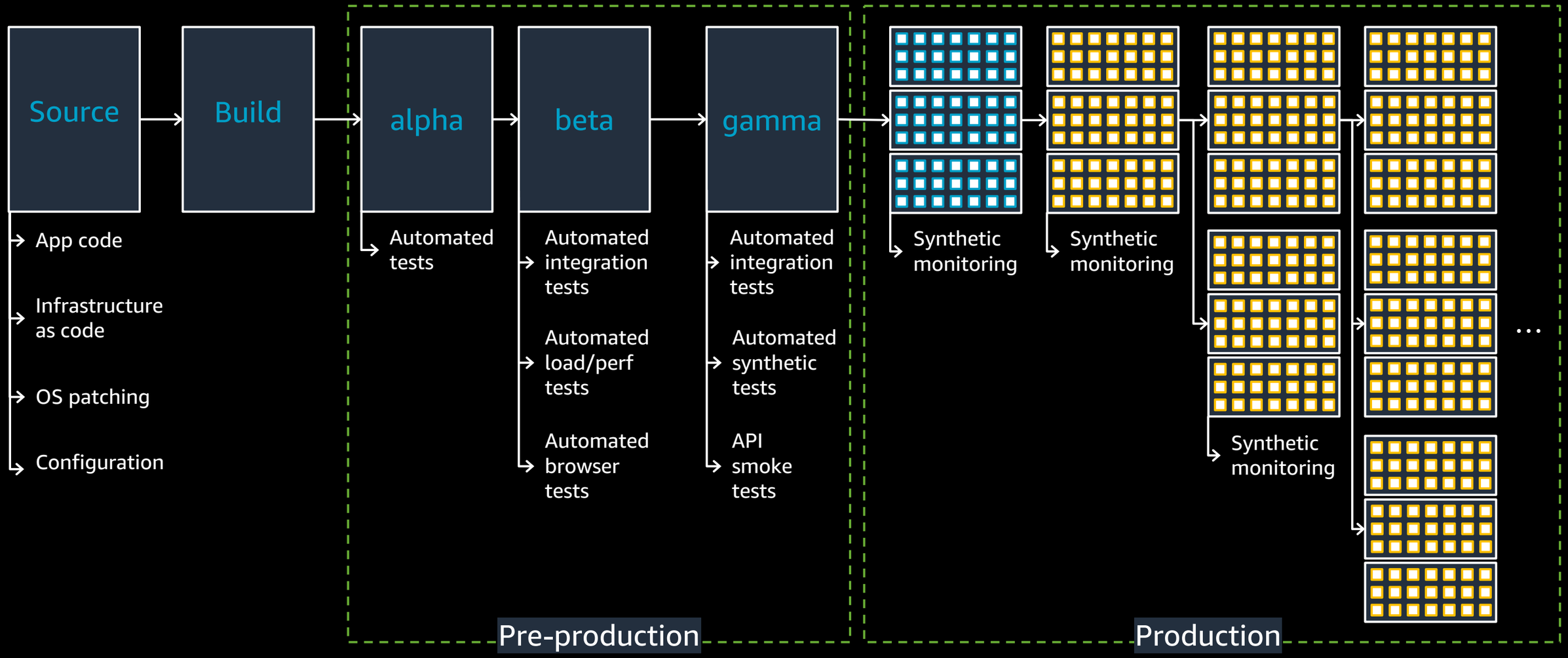
Amazon continuous delivery: Deep dive



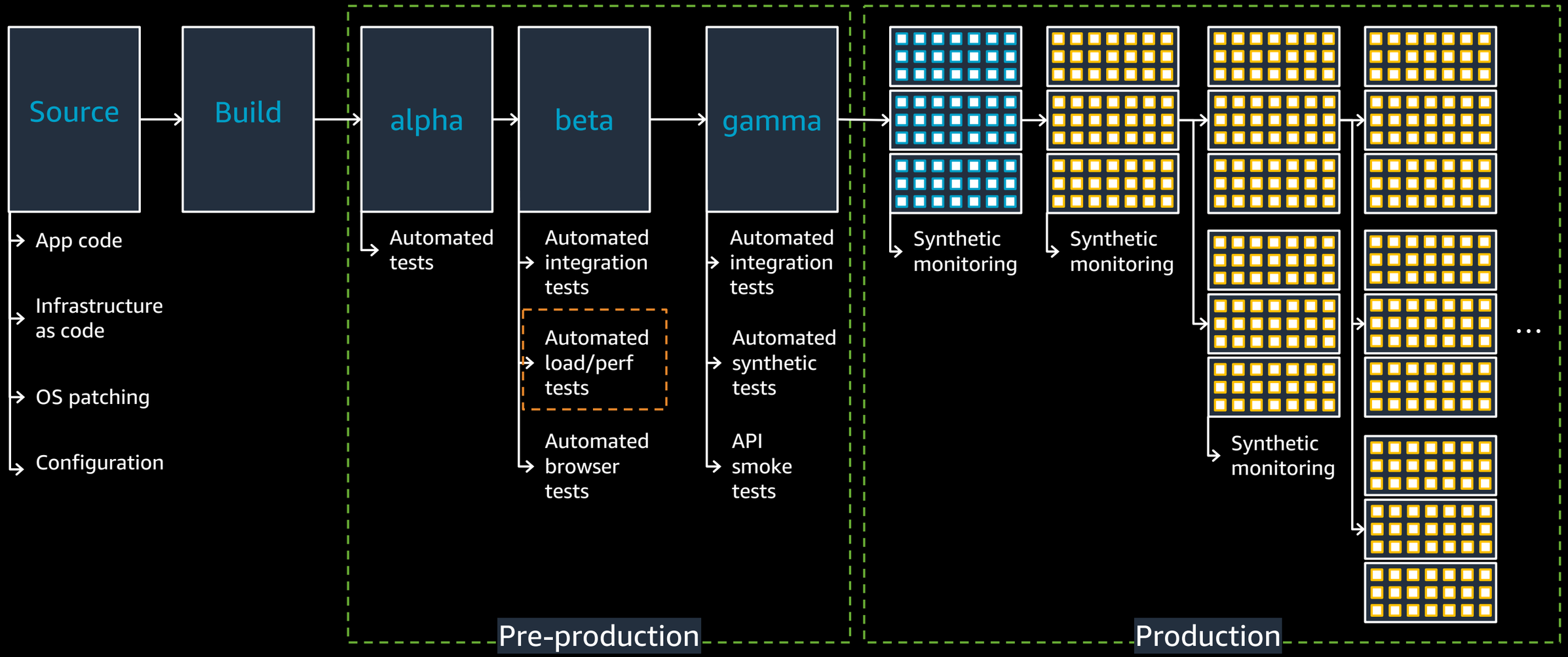
Amazon continuous delivery: Deep dive



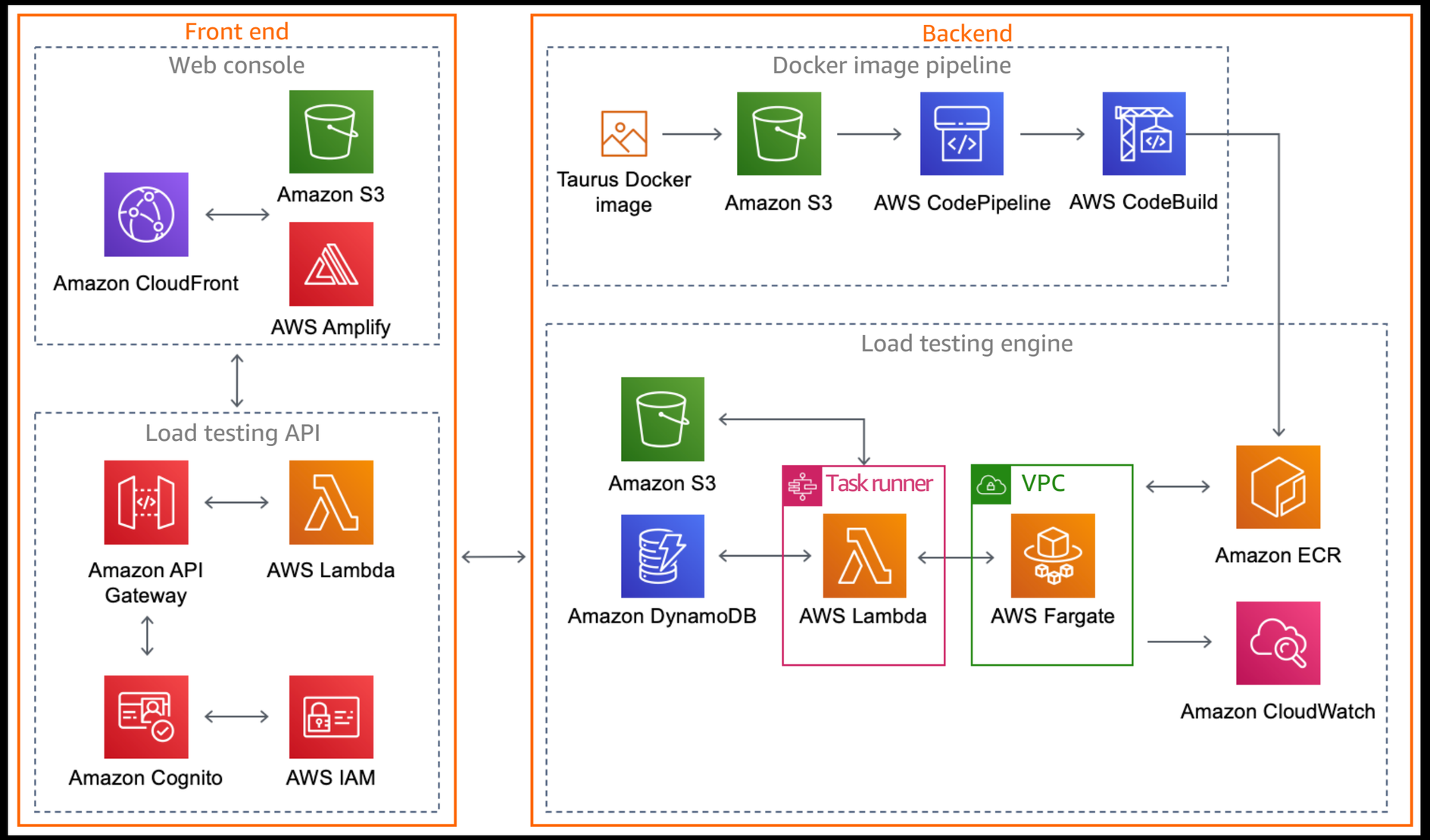
Amazon continuous delivery: Deep dive



Amazon continuous delivery: Deep dive



Distributed load test with Step Functions



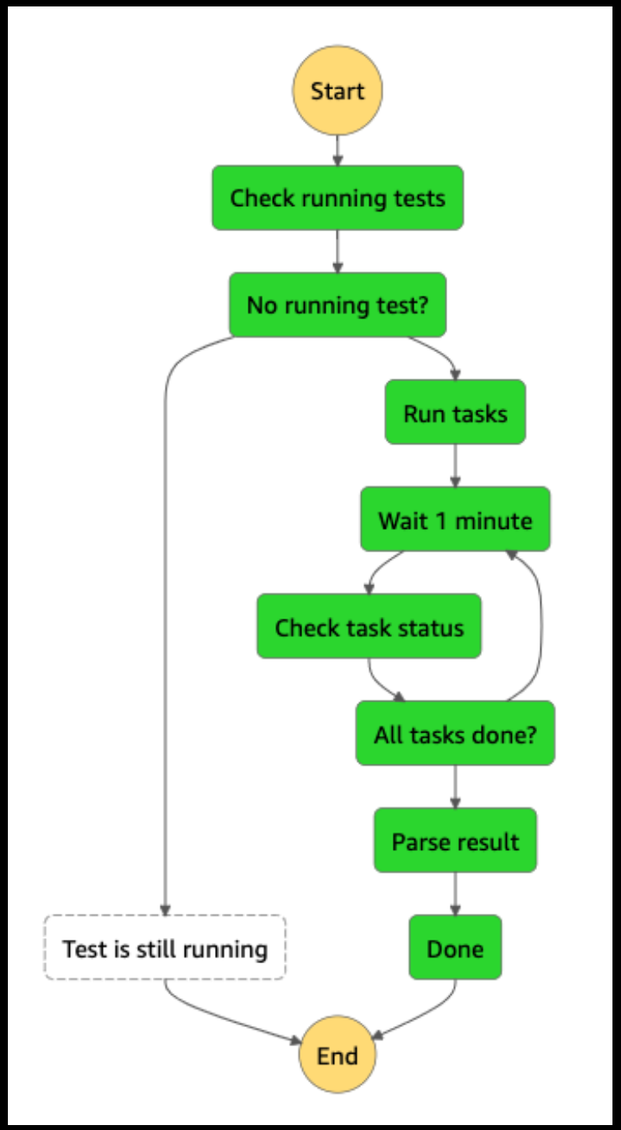
https://aws.amazon.com/solutions/implementations/distributed-load-testing-on-aws/

Distributed load test with Step Functions

- State machine orchestrates distributed load test executed across tasks in ECS Fargate
- Once tasks complete, results are parsed and sent to CloudWatch

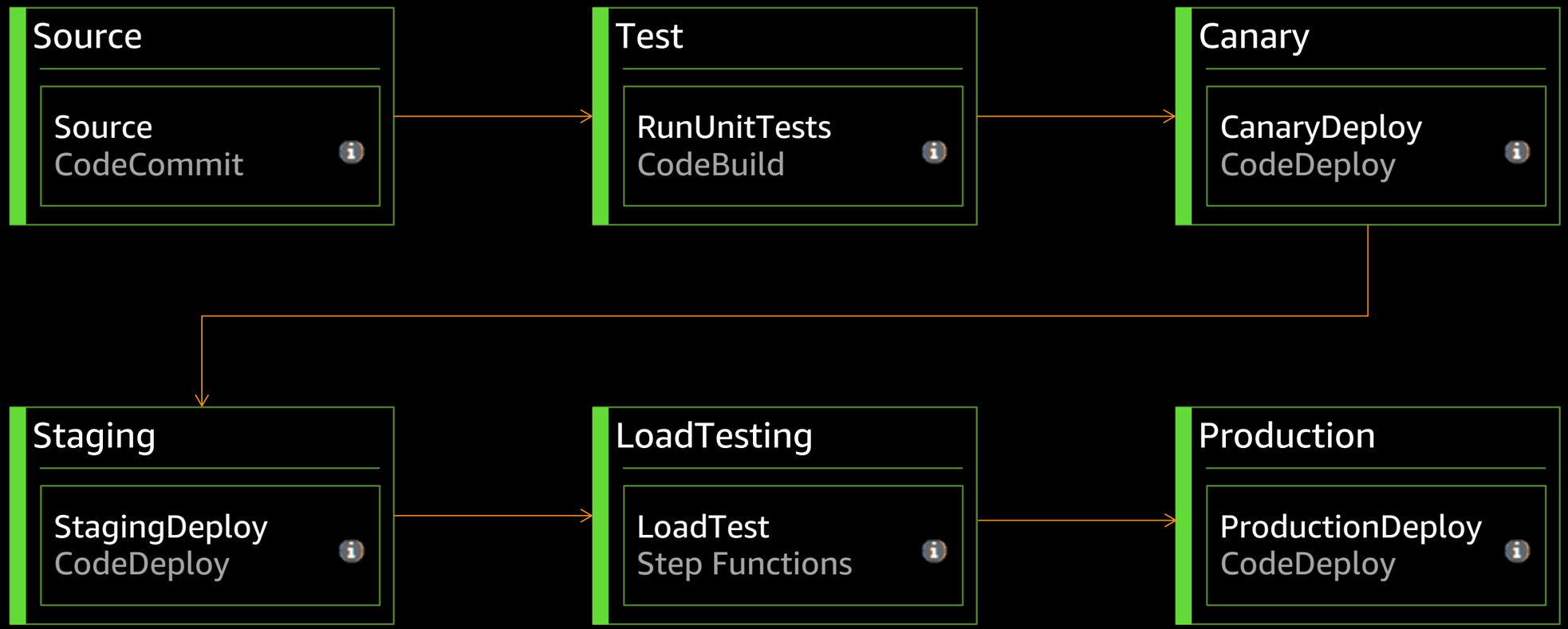
```
"execution": [{
    "concurrency": 200,
    "ramp-up": "0m",
    "hold-for": "3m",
    "scenario": "Leo's Blog"
}],
"scenarios": {
    "Leo's Blog": {
        "requests": [{
            "url": "https://demo.leozh.net",
            "method": "GET",
            "body": {},
            "headers": {}
        }]
    }
},
```

```
{
  "scenario": {
    "testId": "2cPc_8O_Q",
    "taskCount": 10,
    "testType": "simple"
  }
}
```

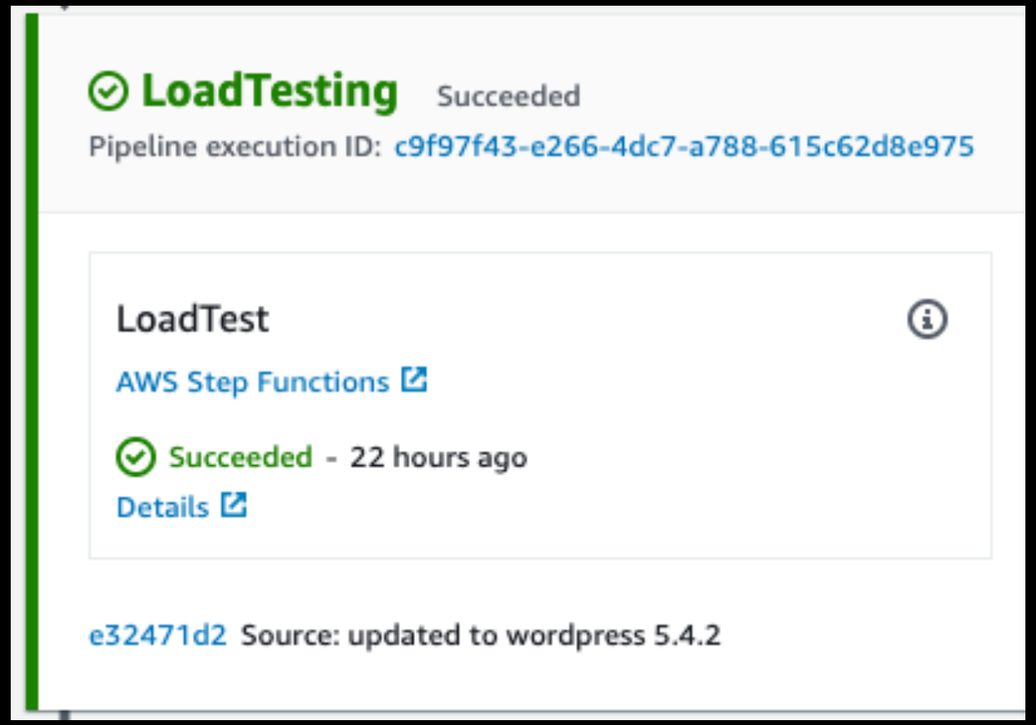


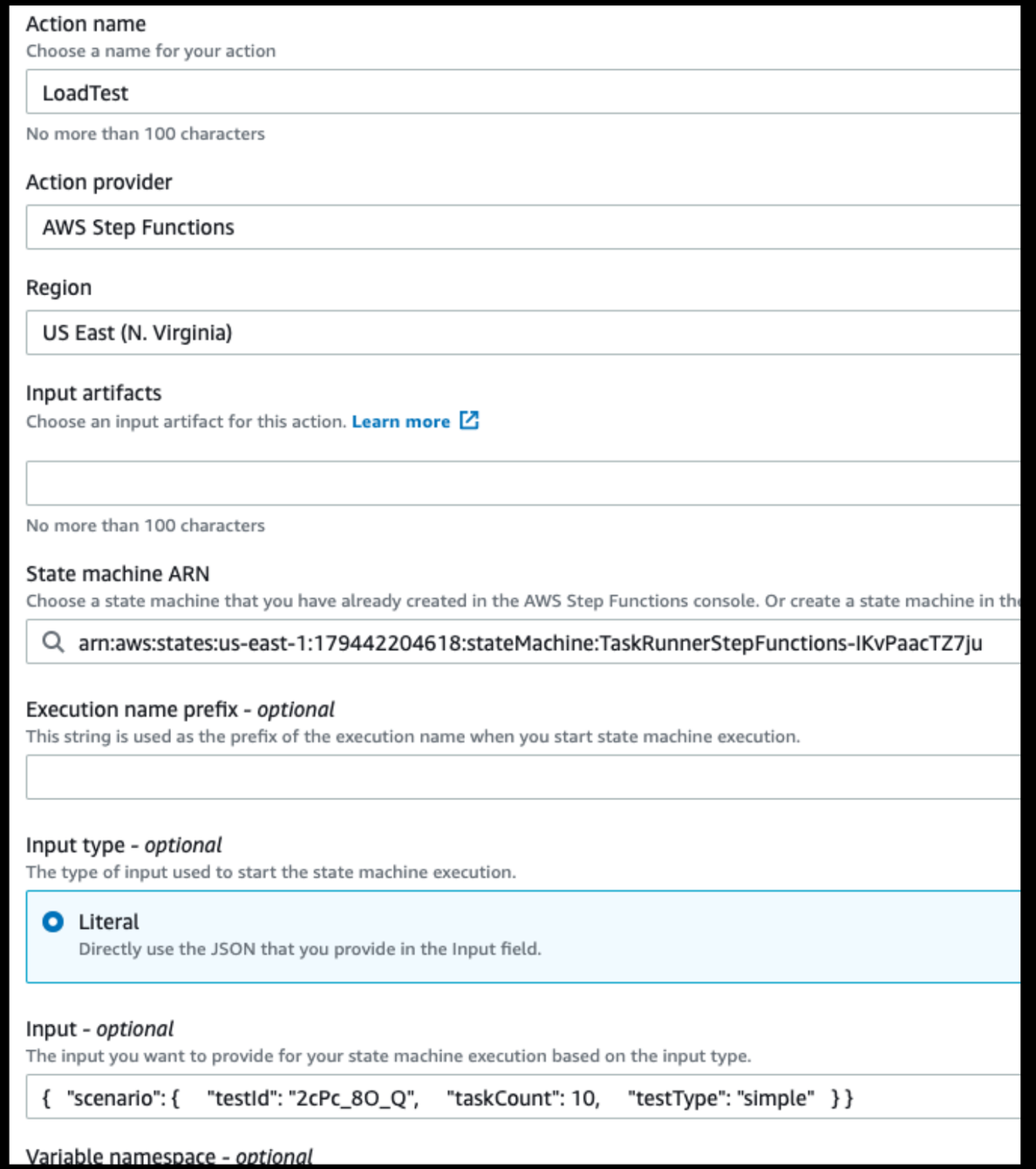
https://aws.amazon.com/solutions/implementations/distributed-load-testing-on-aws/

Putting it all together



Putting it all together





Q&A



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Next steps to learn more

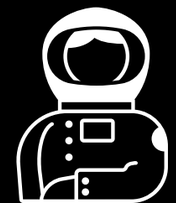
Fine-grained continuous delivery with CodePipeline and AWS Step Functions
Blog post with another CodePipeline and Step Functions example
https://aws.amazon.com/blogs/devops/new-fine-grained-continuous-delivery-with-codepipeline-and-aws-stepfunctions/

Tutorial: Use an AWS Step Functions invoke action in a pipeline
From CodePipeline documentation
https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials-step-functions.html

Sample projects for Step Functions
Examples of Step Functions use cases
https://docs.aws.amazon.com/step-functions/latest/dg/create-sample-projects.html

Distributed load testing on AWS
Distributed load test solution used for demo
https://aws.amazon.com/solutions/implementations/distributed-load-testing-on-aws/

Amazon Builder's Library
Articles and videos about how Amazon builds and operates technologies that underpin Amazon.com and AWS
https://aws.amazon.com/builders-library/?cards-body.sort-by=item.additionalFields.customSort&cards-body.sort-order=asc



Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.