
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

XNT302

# Build, test, and deploy your .NET applications on AWS

Dror Helper (he/him)

Sr. Microsoft Specialist Solutions Architect
AWS

Jagadeesh Chitikesi (he/him)

Sr. Microsoft Specialist Solutions Architect
AWS

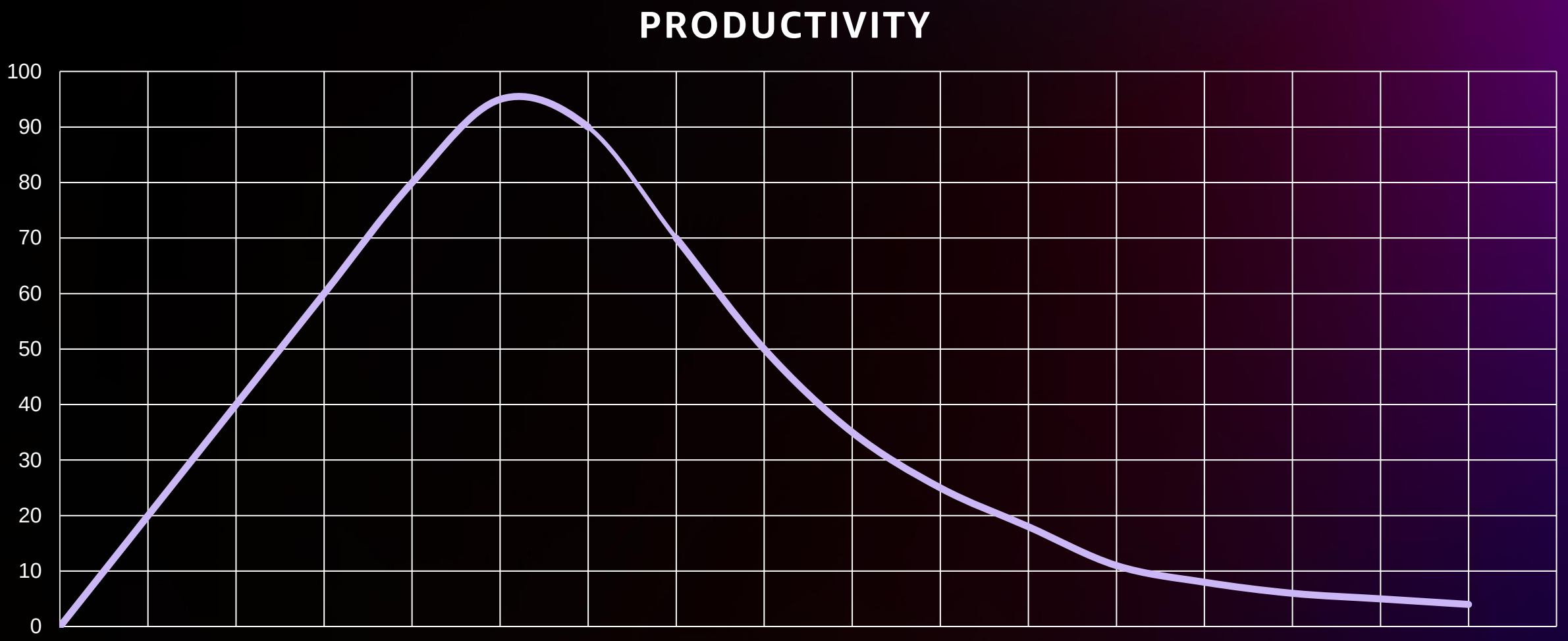
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Typical journey of a greenfield project




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# After a while . . .




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

“Adding manpower to a late software project makes it later.”

**Fred Brooks**

The Mythical Man-Month, 1975



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# How to achieve more with fewer developers?




aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Continuous integration/continuous delivery




aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Continuous integration/continuous delivery

METRICS FROM HIGH-PERFORMING TEAMS AROUND THE WORLD

| | |
|---|---|
| Deployment frequency | On demand (multiple deploys per day) |
| Lead time for changes | Less than one hour |
| Time to restore service | Less than one hour |
| Change failure rate | 0%–15% |

* State of DevOps 2021 report

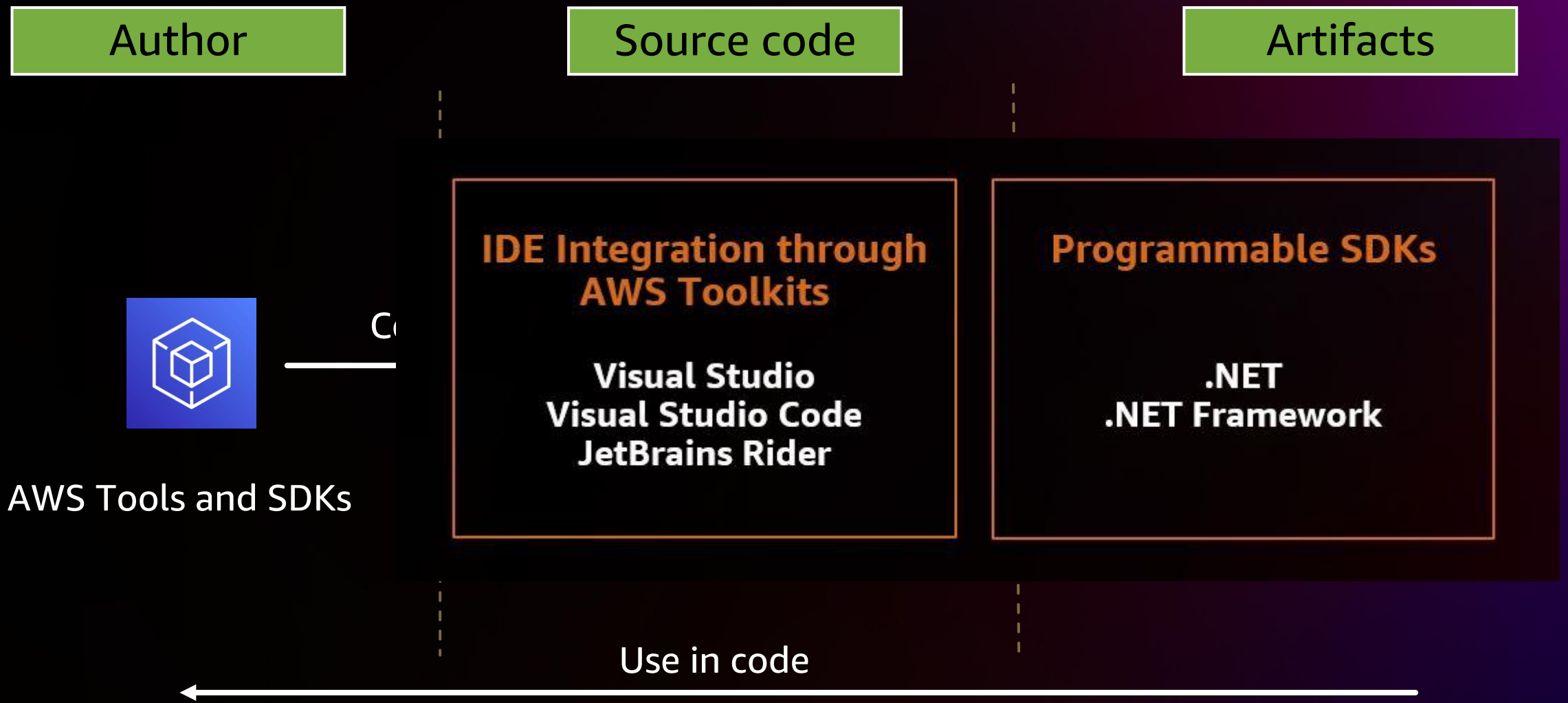
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS source/artifact services




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Automating your build/deploy pipeline

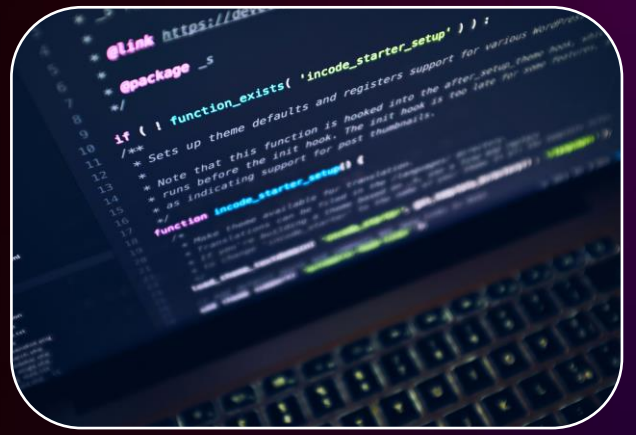
Source → Build → Test → Deploy



Every commit triggers a build



Fix failing builds immediately

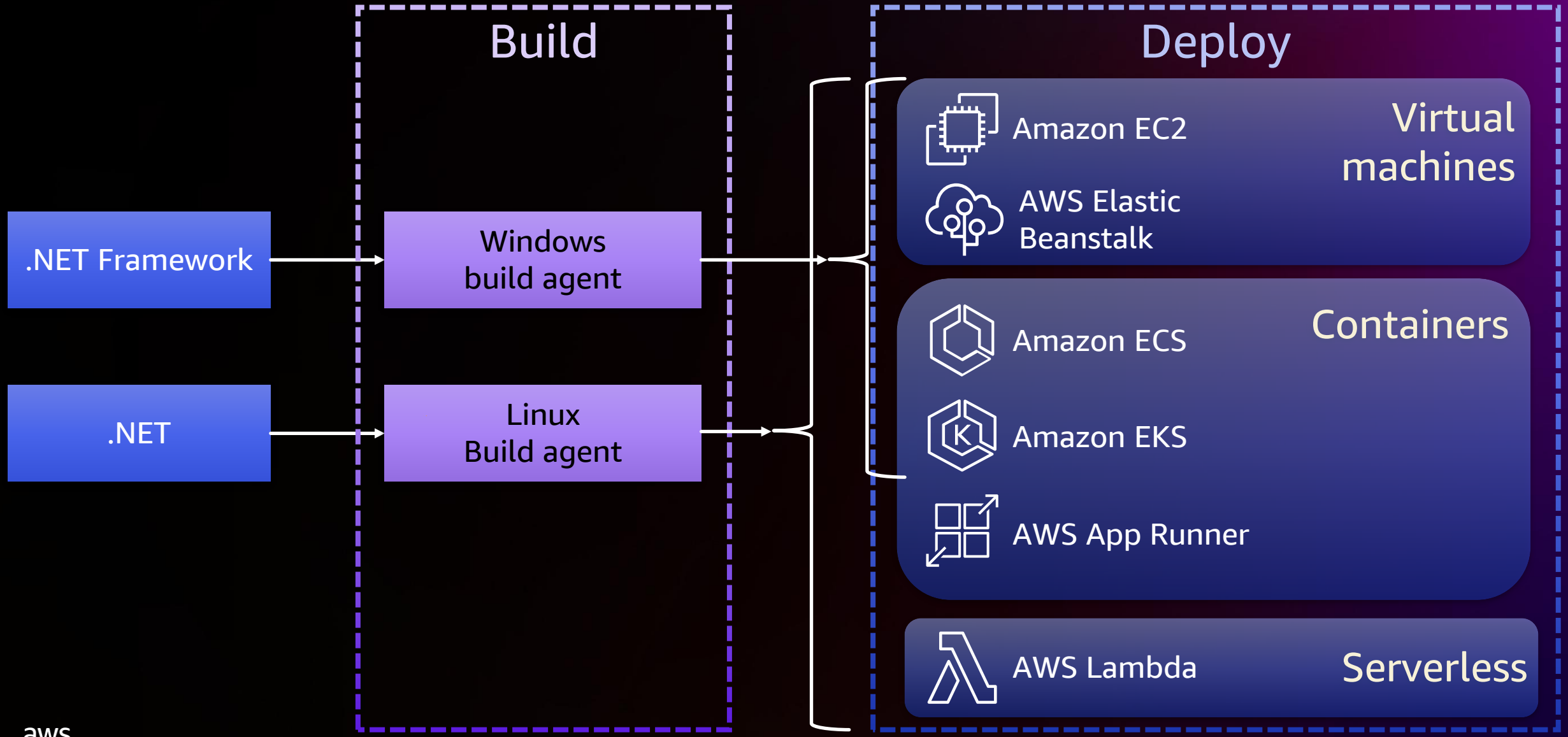


Frequently commit changes

aws

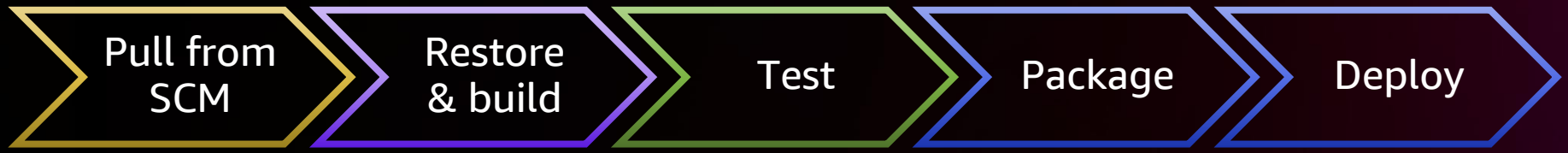
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# .NET build/deploy pipelines




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Deploying an ASP.NET application to IIS





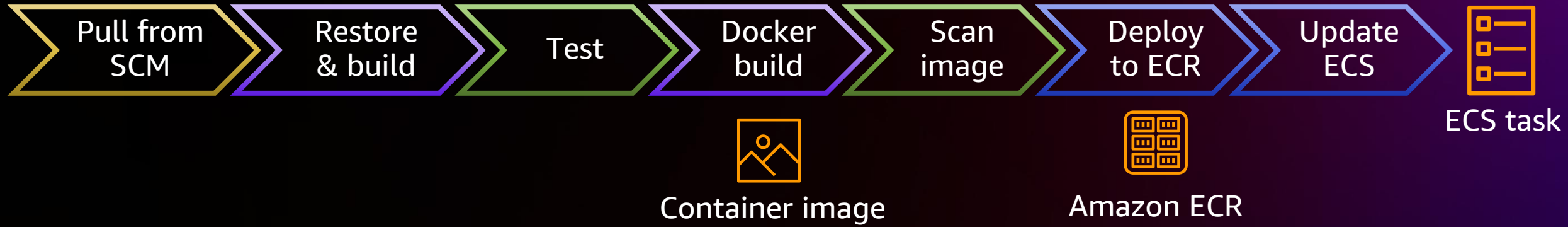

IIS on Amazon EC2



Elastic Beanstalk application

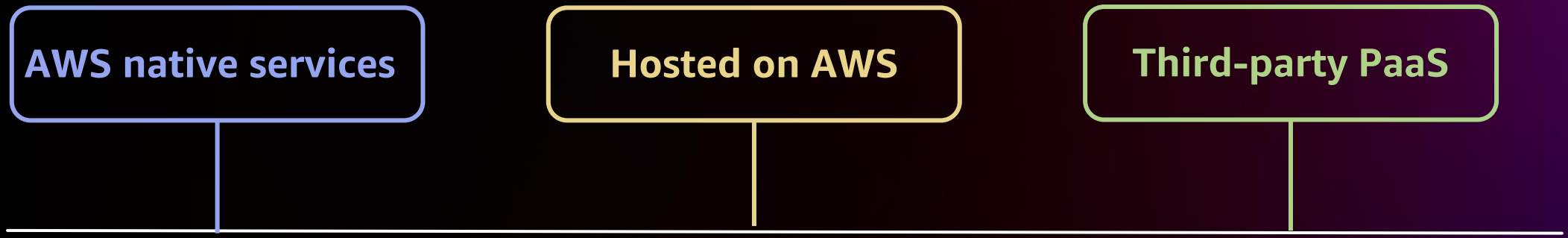
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Deploying a containerized .NET application to Amazon ECS




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Build pipeline tools

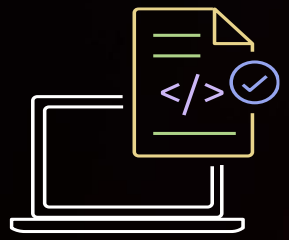



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Choosing your build pipeline tool



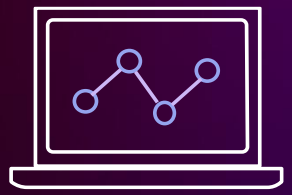
Multiple build environments



Reusable and versioning support



Build visibility and traceability

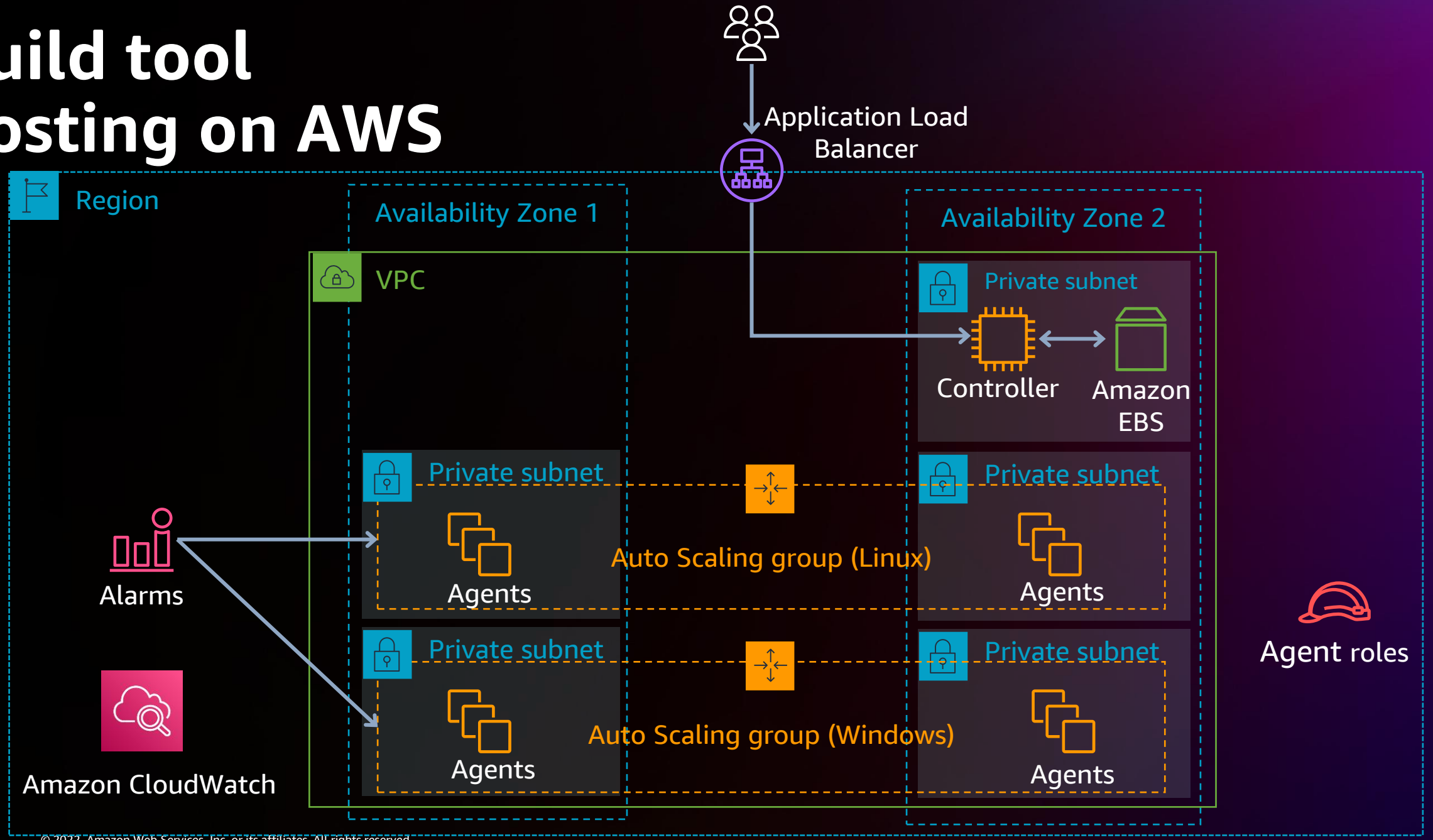


Statistics and data

aws

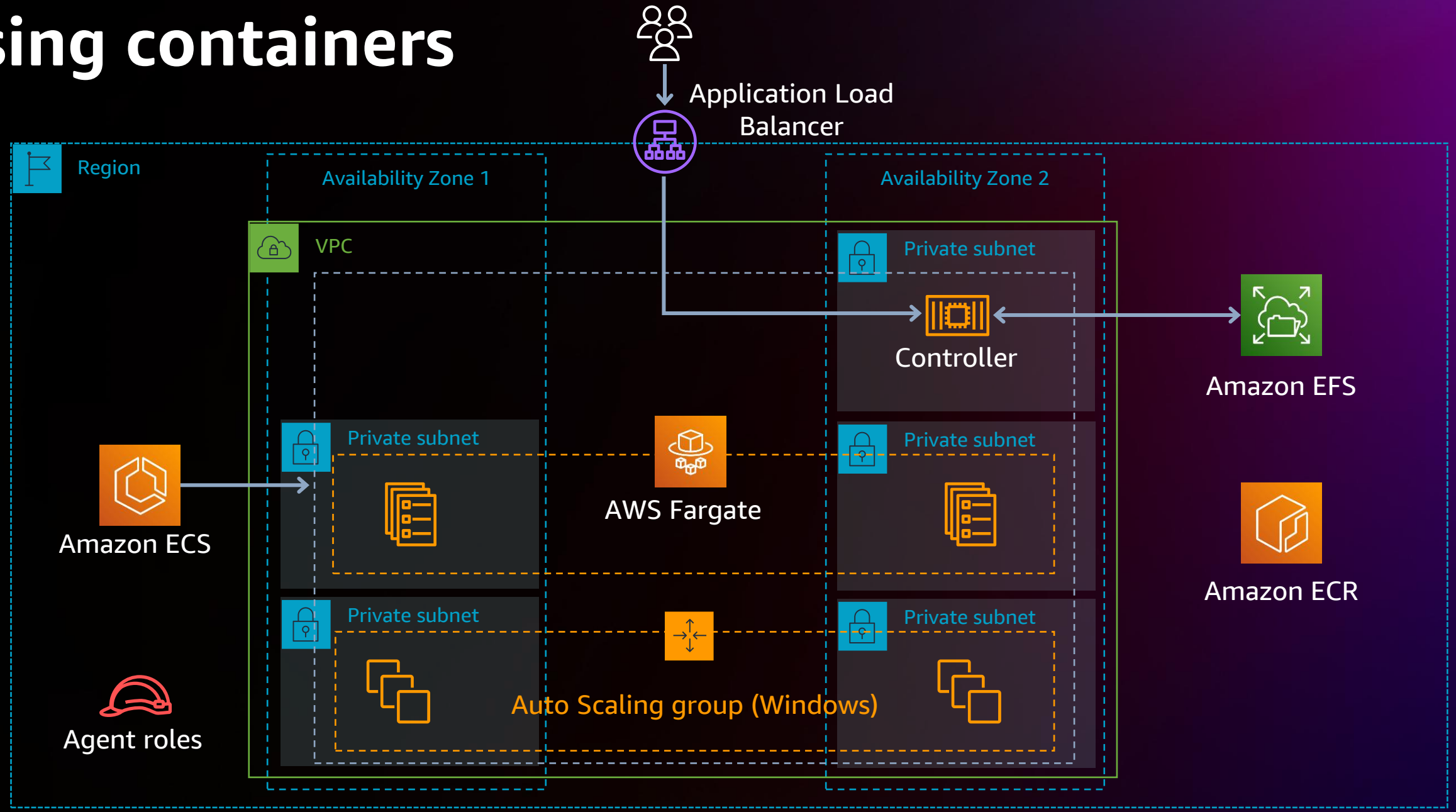
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Build tool hosting on AWS



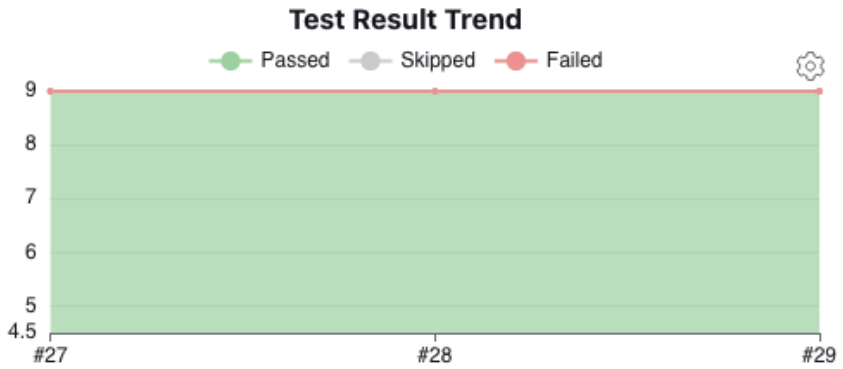

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Using containers




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

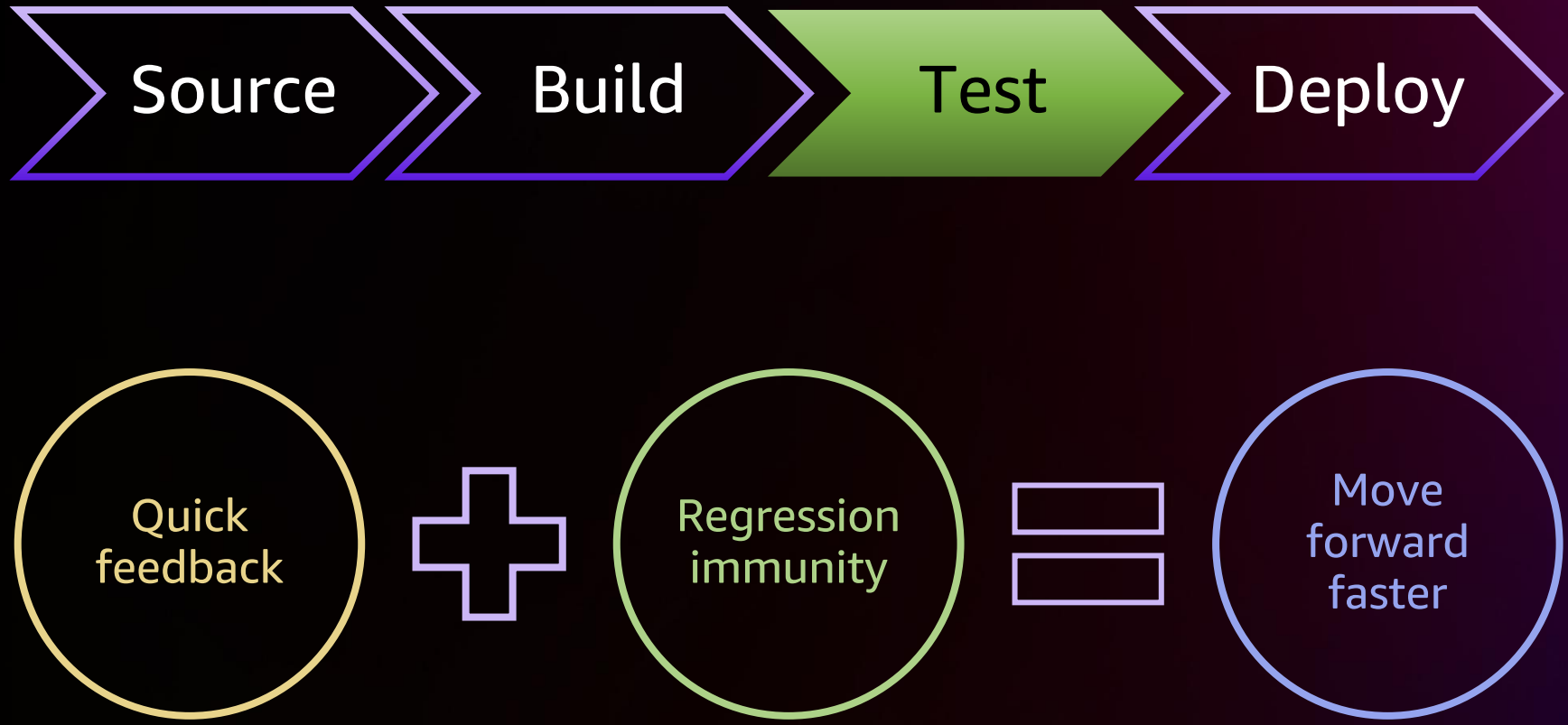
Recent Changes




## Stage View

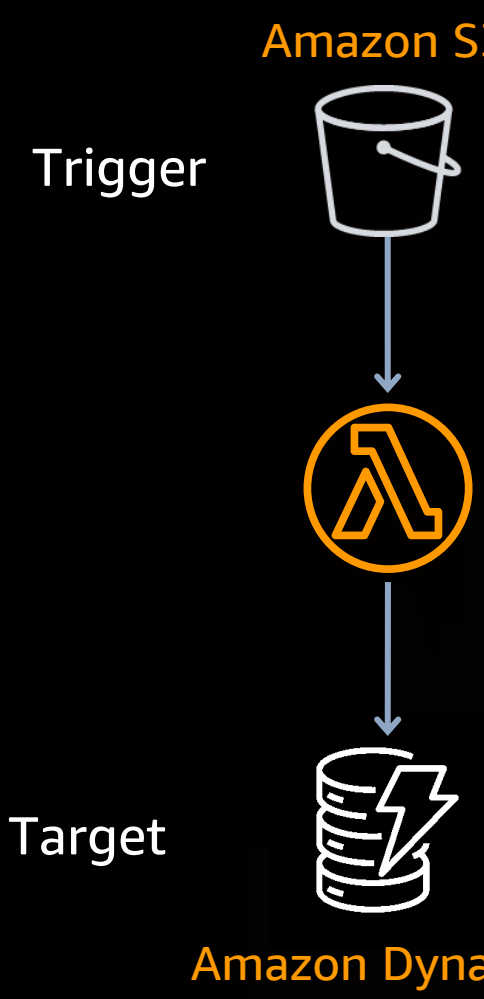
| | Declarative: Checkout SCM | Declarative: Tool Install | Logging into ECR | build | test | Docker build | Scan Docker image | Push to ECR | Trigger ECS update | Declarative: Post Actions |
|---|---|---|---|---|---|---|---|---|---|---|
| Average stage times: (Average full run time: ~25s) | 2s | 212ms | 1s | 3s | 4s | 8s | 518ms | 1s | 794ms | 920ms |
| #29 Jun 21 18:43 — 1 commit | 3s | 175ms | 1s | 2s | 5s | 7s | 481ms | 1s | 1s | 885ms |
| #28 Jun 21 18:19 — 1 commit | 2s | 147ms | 1s | 2s | 5s | 7s | 495ms | 1s | 1s | 825ms |
| #27 Jun 21 18:16 — 2 commits | 3s | 215ms | 1s | 7s | 5s | 8s | 542ms | 1s failed | 40ms failed | 1s |
| #26 Jun 21 18:08 — 3 commits | 2s | 311ms | 1s | 1s | 1s | 7s | 555ms | 1s | 1s | 971ms |

# Test automation is crucial for your build pipeline




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Unit testing Lambda functions




```
public Function()
{
    _s3Client = new AmazonS3Client();
    _dynamoDbClient = new AmazonDynamoDBClient();
}
```

```
public
Function(IAmazonS3 s3Client, IAmazonDynamoDB dynamoDbClient)
{
    _s3Client = s3Client;
    _dynamoDbClient = dynamoDbClient;
}
```

```
public async Task<string?>
FunctionHandler(S3Event evnt, ILambdaContext ctx)
{
    . . .
}
```

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# How to unit test Lambda function




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Unit testing Lambda function

```
[Fact]
public async Task Invoke_S3EventEmpty_returnWithoutCallingDynamoDB()
{
    var fakeS3Client = A.Fake<IAmazonS3>();
    var fakeDynamoDbClient = A.Fake<IAmazonDynamoDB>();

    var testContext = new TestLambdaContext();
    var s3Event = new S3Event();
    var lambda = new Function(fakeS3Client, fakeDynamoDbClient);

    await lambda.FunctionHandler(s3Event, testContext);


    A.CallTo(() =>
      fakeDynamoDbClient.PutItemAsync(. . .)).MustNotHaveHappened();
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Unit testing Lambda function

```
[Fact]
public async Task Invoke_S3EventEmpty_returnWithoutCallingDynamoDB()
{
    var fakeS3Client = A.Fake<IAmazonS3>();
    var fakeDynamoDbClient = A.Fake<IAmazonDynamoDB>();

    var testContext = new TestLambdaContext();
    var s3Event = new S3Event();
    var lambda = new Function(fakeS3Client, fakeDynamoDbClient);

    await lambda.FunctionHandler(s3Event, testContext);


    A.CallTo(() =>
      fakeDynamoDbClient.PutItemAsync(. . .)).MustNotHaveHappened();
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Unit testing Lambda function

```
[Fact]
public async Task Invoke_S3EventEmpty_returnWithoutCallingDynamoDB()
{
    var fakeS3Client = A.Fake<IAmazonS3>();
    var fakeDynamoDbClient = A.Fake<IAmazonDynamoDB>();

    var testContext = new TestLambdaContext();
    var s3Event = new S3Event();
    var lambda = new Function(fakeS3Client, fakeDynamoDbClient);

    await lambda.FunctionHandler(s3Event, testContext);


    A.CallTo(() =>
      fakeDynamoDbClient.PutItemAsync(. . .)).MustNotHaveHappened();
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Unit testing Lambda function

```
[Fact]
public async Task Invoke_S3EventEmpty_returnWithoutCallingDynamoDB()
{
    var fakeS3Client = A.Fake<IAmazonS3>();
    var fakeDynamoDbClient = A.Fake<IAmazonDynamoDB>();

    var testContext = new TestLambdaContext();
    var s3Event = new S3Event();
    var lambda = new Function(fakeS3Client, fakeDynamoDbClient);

    await lambda.FunctionHandler(s3Event, testContext);


    A.CallTo(() =>
      fakeDynamoDbClient.PutItemAsync(. . .)).MustNotHaveHappened();
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Unit testing Lambda function

```
[Fact]
public async Task Invoke_S3EventEmpty_returnWithoutCallingDynamoDB()
{
    var fakeS3Client = A.Fake<IAmazonS3>();
    var fakeDynamoDbClient = A.Fake<IAmazonDynamoDB>();

    var testContext = new TestLambdaContext();
    var s3Event = new S3Event();
    var lambda = new Function(fakeS3Client, fakeDynamoDbClient);

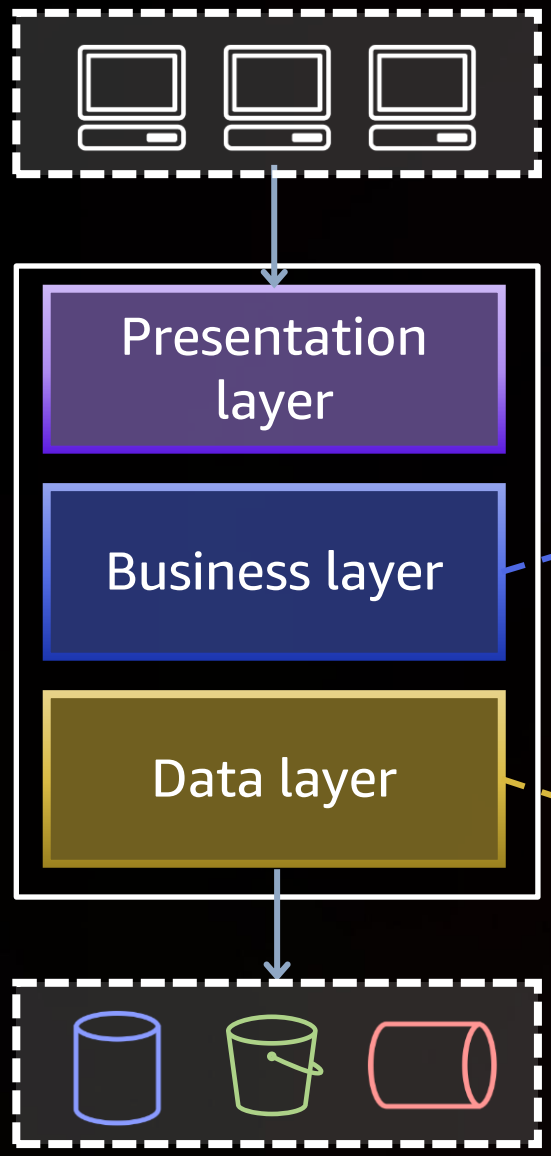
    await lambda.FunctionHandler(s3Event, testContext);

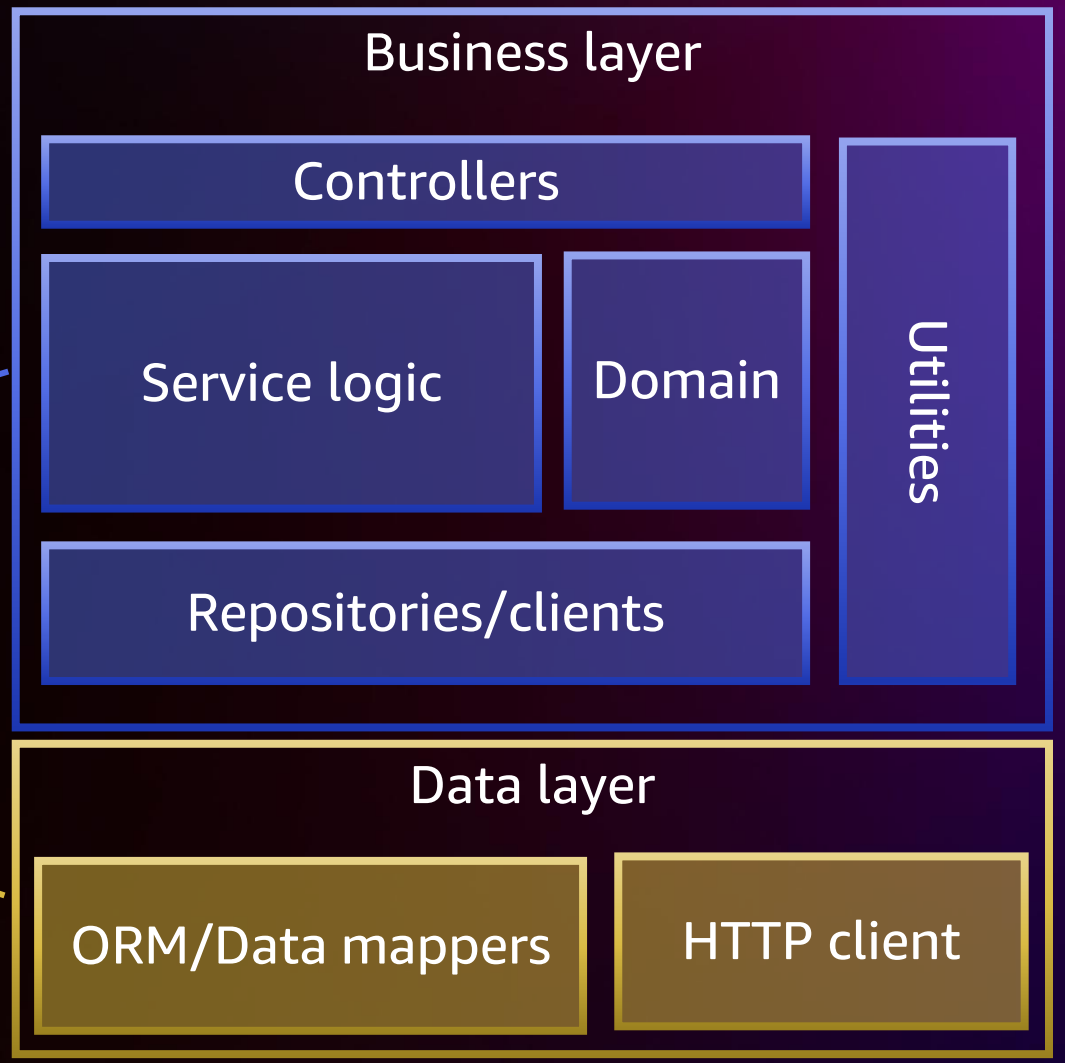

    A.CallTo(() =>
      fakeDynamoDbClient.PutItemAsync(. . .)).MustNotHaveHappened();
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

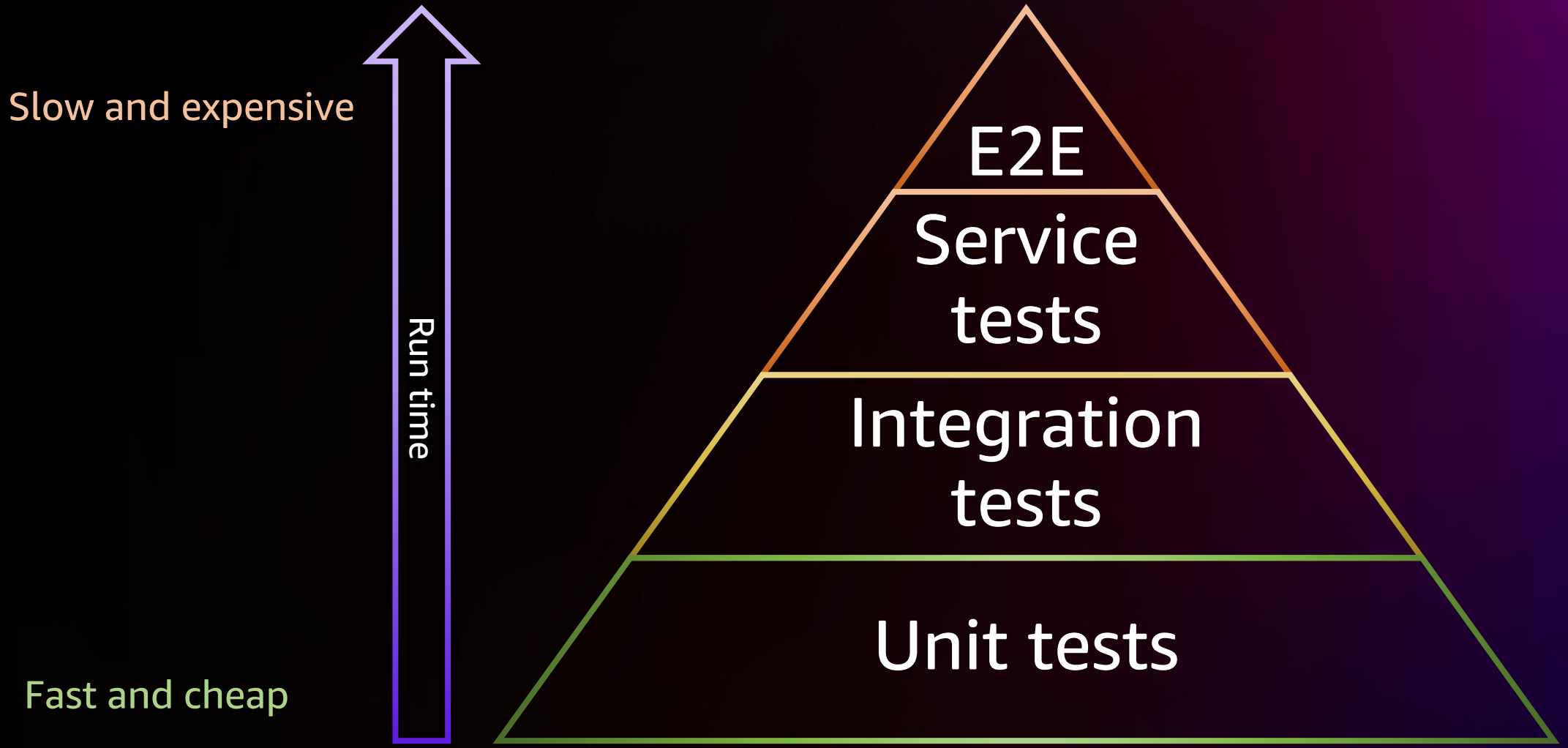
# ASP.NET microservice/service structure




aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# The test pyramid




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Infrastructure as code (IaC)

## A key DevOps methodology to define and deploy infrastructure using code

- Repeatable and consistent
- Code reviews
- Speed and visibility



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Infrastructure as code (IaC)

CHOOSE THE BEST TOOL FOR YOU

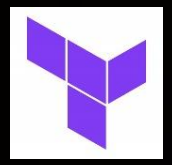
Op  AWS CloudFormation

Dev 

AWS Serverless Application Model (AWS SAM)

Dev  AWS Cloud Development Kit (AWS CDK)

---

Op  Terraform

Dev  Pulumi

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS CDK

DEFINE AND DEPLOY YOUR AWS INFRASTRUCTURE USING THE FAMILIARITY OF YOUR FAVORITE PROGRAMMING LANGUAGE



Use C# to define  resources



IntelliSense
Inline Documentation



Smart defaults
Reusable constructs



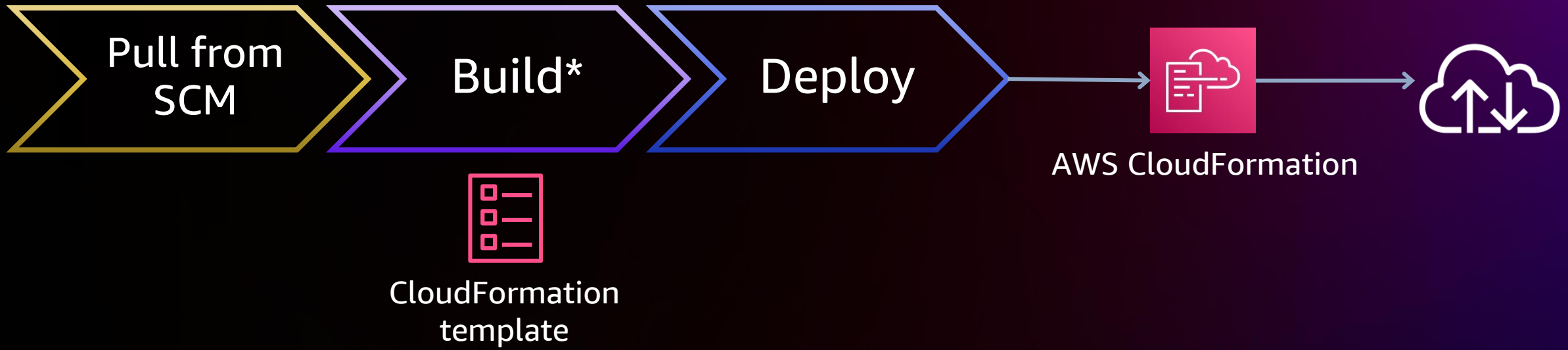
Compile time error detection




aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# IaC build pipeline




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Tying it all together

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Thank you!



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.