
AWS Certified Developer - Associate

Release 1.0

Jose Antonio Alvarez Cubero

Dec 19, 2020

Contents

1	Contents:	1
1.1	Preface	1
1.2	Developer Tools Services	1
1.3	Deployment as Code	2
2	Indices and tables	5

1.1 Preface

1.1.1 Authors

- Jose Antonio Alvarez Cubero

Who Should Read This Book?

Expected Audience

The intended audience for this book is those with a general need to understand main AWS services. While AWS certification candidates may experience the largest benefit from this content, the materials included herein may be of use to a much wider audience, especially given modern industry trends towards AWS cloud computing technologies.

There are many elements in this book that explore topics outside the typical contents of AWS certifications. For example the use of AWS python SDK can be viewed as a development-oriented task, however has specific relevance to cloud computing configuration and function, taking a very different approach than traditional CLI-based interface configuration.

Organization of this Book

1.2 Developer Tools Services

1.2.1 AWS Tools & SDKs

[Boto 3 Documentation](#)

1.2.2 AWS X-Ray

1.2.3 Amazon Honeycode

Amazon Honeycode

1.3 Deployment as Code

1.3.1 AWS CodePipeline

1.3.2 AWS CodeCommit

1.3.3 AWS CodeBuild

1.3.4 AWS CodeDeploy

CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services. It allows you to rapidly release new features, update Lambda function versions, avoid downtime during application deployment, and handle the complexity of updating your applications, without many of the risks associated with error-prone manual deployments.

Deployment Groups

You can specify one or more deployment groups for a CodeDeploy application. Each application deployment uses one of its deployment groups. The deployment group contains settings and configurations used during the deployment. Most deployment group settings depend on the compute platform used by your application. Some settings, such as rollbacks, triggers, and alarms can be configured for deployment groups for any compute platform.

Deployment type options

CodeDeploy provides two deployment type options: In-place deployment, Blue/green deployment.

In-place deployment

The application on each instance in the deployment group is stopped, the latest application revision is installed, and the new version of the application is started and validated. You can use a load balancer so that each instance is deregistered during its deployment and then restored to service after the deployment is complete. Only deployments that use the EC2/On-Premises compute platform can use in-place deployments.

Note: AWS Lambda and Amazon ECS deployments cannot use an in-place deployment type.

Blue/green deployment

A blue/green deployment reroutes traffic from your application's original environment to a replacement environment. Your environment depends on your CodeDeploy application's compute platform.

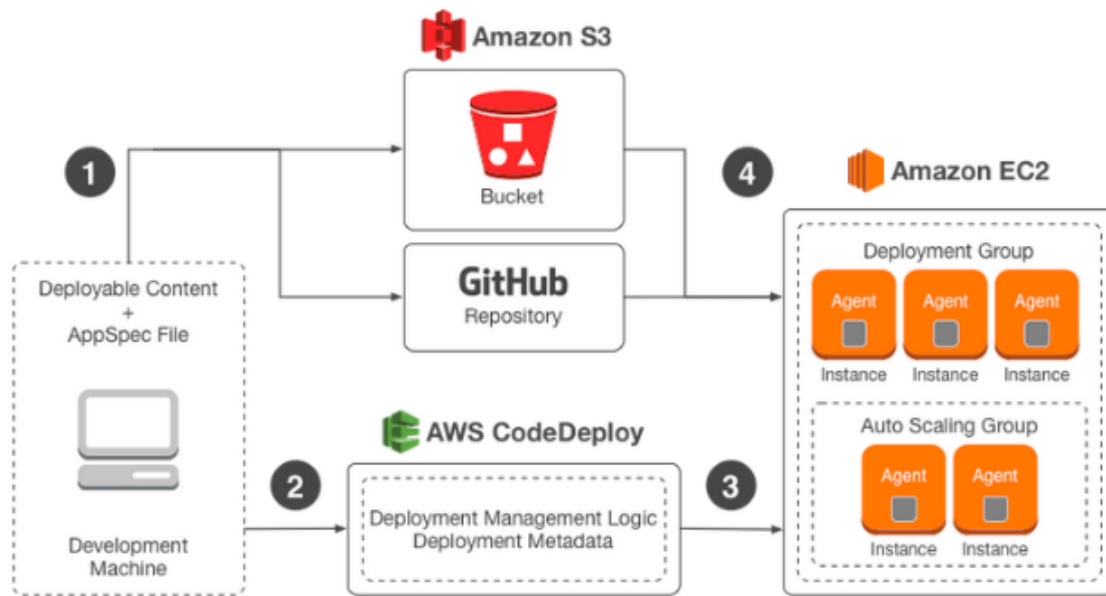


Fig. 1: Deployment Groups

- **AWS Lambda:** Traffic is shifted from one version of a Lambda function to a new version of the same Lambda function.

You must choose one of the following deployment configuration types to specify how traffic is shifted from the original AWS Lambda function version to the new AWS Lambda function version:

- **Canary:** Traffic is shifted in two increments. You can choose from predefined canary options that specify the percentage of traffic shifted to your updated Lambda function version in the first increment and the interval, in minutes, before the remaining traffic is shifted in the second increment.
- **Linear:** Traffic is shifted in equal increments with an equal number of minutes between each increment. You can choose from predefined linear options that specify the percentage of traffic shifted in each increment and the number of minutes between each increment.
- **All-at-once:** All traffic is shifted from the original Lambda function to the updated Lambda function version all at once.
- **Amazon ECS:** Traffic is shifted from a task set in your Amazon ECS service to an updated, replacement task set in the same Amazon ECS service.
- **EC2/On-Premises:** Traffic is shifted from one set of instances in the original environment to a replacement set of instances.

The advantages of using Blue/Green deployment over in-place upgrade strategy are:

Traditionally, with in-place upgrades, it was difficult to validate your new application version in a production deployment while also continuing to run your old version of the application. Blue/green deployments provide a level of isolation between your blue and green application environments. It ensures that spinning up a parallel green environment does not affect resources underpinning your blue environment. This isolation reduces your deployment risk.

After you deploy the green environment, you have the opportunity to validate it. You might do that with test traffic before sending production traffic to the green environment, or by using a very small fraction of production traffic, to better reflect real user traffic. This is called canary analysis or canary testing. If you discover the green environment is not operating as expected, there is no impact on the blue environment. You can route traffic back to it, minimizing impaired operation or downtime, and limiting the blast radius of impact.

This ability to simply roll traffic back to the still-operating blue environment is a key benefit of blue/green deployments. You can roll back to the blue environment at any time during the deployment process. Blue/green deployments also fit well with continuous integration and continuous deployment (CI/CD) workflows, in many cases limiting their complexity. Your deployment automation would have to consider fewer dependencies on an existing environment, state, or configuration.

In AWS, blue/green deployments also provide cost optimization benefits. You're not tied to the same underlying resources. So if the performance envelope of the application changes from one version to another, you simply launch the new environment with optimized resources, whether that means fewer resources or just different compute resources. You also don't have to run an overprovisioned architecture for an extended period of time.

Blue/Green Deployments on AWS

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`