

AWS - CodeDeploy

By

Keshav Kummari

AWS - CodeDeploy

- CodeDeploy – Automates deployments of your applications to EC2, Lambda, and even on-premises environments.
- Deploy your applications in a fast, consistent, and reliable way.
- Benefits of CodeDeploy:
 - Automated Deployments
 - Minimize Downtime
 - Stop the Rollback
 - Centralized Control

Deployment Types

- In-place deployment:
 - The existing servers are updated with the new version of an application
- Blue/Green deployment(EC2):
 - New application versions are deployed on a new set of instances
 - Traffic is routed from old to new instances
 - If there are failures, the application can fall back to the older deployment version
- Blue/Green Deployment(Lambda):
 - Traffic is shifted from one lambda version to another, this can happen in multiple ways:

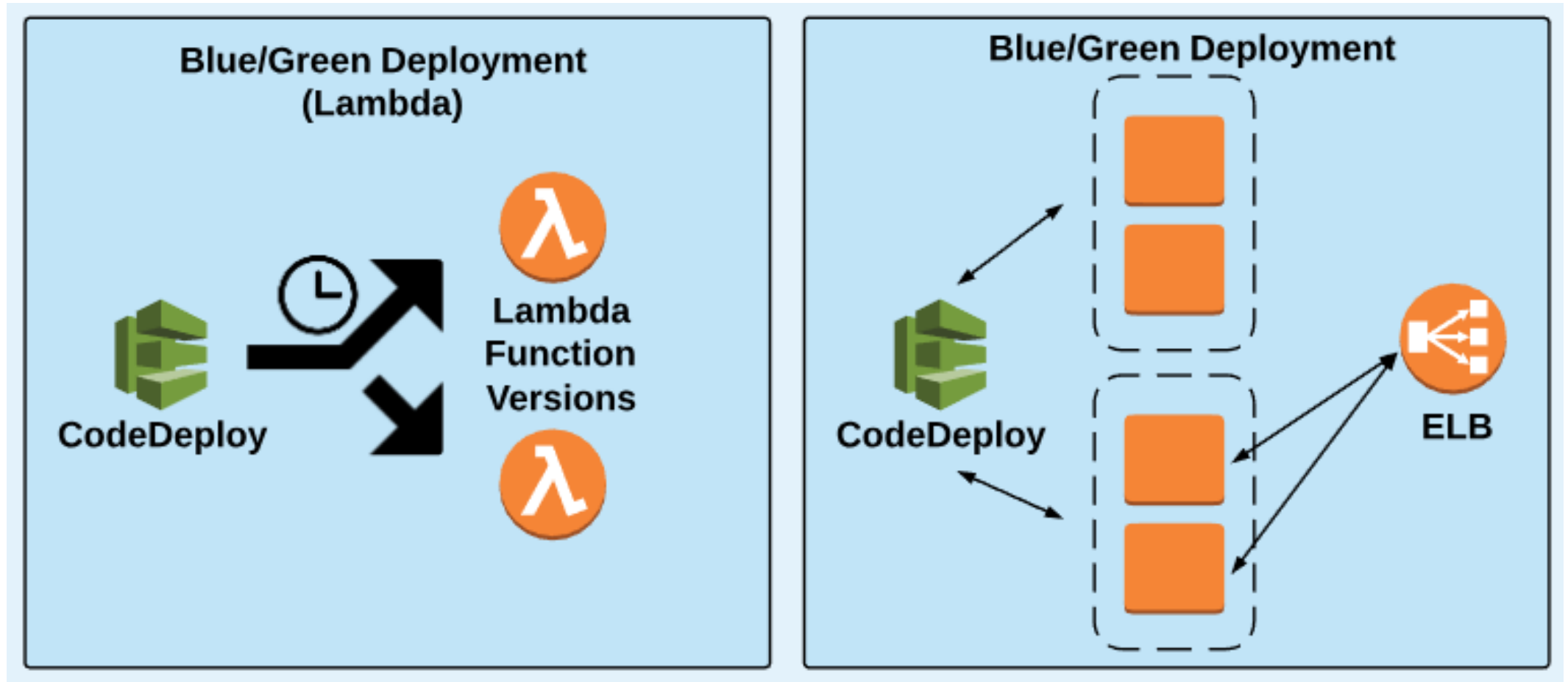
- Canary:
 - A percentage of traffic is shifted to the new version
 - CodeDeploy then waits for a specified time and shifts the rest of the traffic if it sees no errors.
- Linear:
 - Traffic is shifted in equal increments with an equal number of minutes between each increment.
- All at once:
 - Traffic is immediately and completely shifted to the new version of the Lambda function.

CodeDeploy Hooks

- CodeDeploy AppSpec File:
 - Lambda Deployments:
 - Which functions to deploy
 - Which functions to use as validation tests
 - EC2/On-Premises Deployments:
 - What to install (From S3 or Github)
 - What lifecycle events hooks to run in response to deployment lifecycle events
- Lifecycle Hooks:
 - Available hooks depend on the deployment type
 - Hooks allow arbitrary scripts to run during your deployment process
 - Typical examples include:
 - BeforeInstall
 - AfterInstall
 - ApplicationStart
 - ApplicationStop
 - ValidateService

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/WordPress
hooks:
  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root
  AfterInstall:
    - location: scripts/change_permissions.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_server.sh
    - location: scripts/create_test_db.sh
      timeout: 300
      runas: root
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root
```

CodeDeploy Examples



Click on “Sample Deployment” and click on “Next”

Step 1: Get started

Step 2: Choose a deployment type

Get started with AWS CodeDeploy

AWS CodeDeploy helps you to quickly deploy applications to Amazon EC2 instances or on-premises instances.

Start by creating a deployment that uses a sample application supplied by AWS CodeDeploy, or skip this wizard and create a custom deployment with your own application.



Sample deployment

Recommended for new AWS CodeDeploy users.



Custom deployment

Recommended if you already have instances and an application to deploy.

Cancel

Next

Let's select "Blue/Green Deployment"

Step 1: Get started

Step 2: Choose a deployment type

Step 3: Create blue/green deployment

Choose a deployment type

Choose the deployment to use to deploy your application.



Blue/green deployment

Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.



In-place deployment

Updates the instances in the deployment group with the latest application revision. During a deployment, each instance will be briefly taken offline for its update.

[Cancel](#)

[Previous](#)

[Next](#)

Fill the required fields and click on “Launch environment”

[Step 1: Get started](#)

[Step 2: Choose a deployment type](#)

Step 3: Create blue/green deployment

Create blue/green deployment ?

We will launch an environment using the following configuration for you to try a blue/green deployment.

Application name* ⓘ

Deployment group name* ⓘ

Auto Scaling group name* ⓘ

Load balancer name* ⓘ

Service role name* ⓘ

Key pair name* ⓘ

After the environment is launched, it can be used in a blue/green deployment.

Launch environment

*Required

[Cancel](#)

[Previous](#)

[Start blue/green deployment](#)

It's under process...

 We're setting up your environment now. It **might take a few minutes.**

Sample environment:

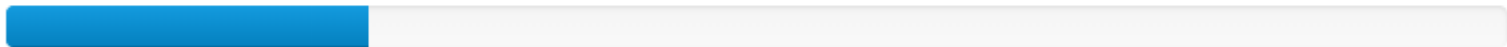
We are launching a Classic load balancer (BlueGreenLoadBalancer) and an Auto Scaling group (BlueGreenAutoScalingGroup) of three t2.micro EC2 instances.

Sample application:

After your environment is launched, we will install a sample application using an AWS CodeDeploy in-place deployment.

You can download the application code:

https://s3.ap-south-1.amazonaws.com/aws-codedeploy-ap-south-1/samples/latest/SampleApp_Linux.zip



Step 14 of 58 complete

[See more details in AWS CloudFormation](#)

*Required

Cancel

Previous

Start blue/green deployment

CloudFormation is under process..

CloudFormation

Stacks

Create Stack

Actions

Design template

Filter: Active

By Stack Name

Showing 1 stack

	Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/>	CodeDeployBGStack-5ee03gy	2018-07-06 19:08:41 UTC+0550	CREATE_IN_PROGRE...	This template creates the resources required for a sample AWS CodeDeploy blue/green ...

Overview

Outputs

Resources

Events

Template

Parameters

Tags

Stack Policy

Change Sets

Rollback Triggers

Filter by: Status

Search events

2018-07-06	Status	Type	Logical ID	Status Reason
▶ 19:09:40 UTC+0550	CREATE_COMPLETE	AWS::EC2::SubnetRouteTableAssociation	PublicSubnetRouteTableAssociation	
▶ 19:09:26 UTC+0550	CREATE_COMPLETE	AWS::ElasticLoadBalancing::LoadBalancer	ClassicLoadBalancer	
▶ 19:09:25 UTC+0550	CREATE_IN_PROGRESS	AWS::ElasticLoadBalancing::LoadBalancer	ClassicLoadBalancer	Resource creation Initiated
▶ 19:09:24 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	PublicSubnetRouteTableAssociation	Resource creation Initiated
19:09:24 UTC+0550	CREATE_IN_PROGRESS	AWS::ElasticLoadBalancing::LoadBalancer	ClassicLoadBalancer	
19:09:24 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::SubnetRouteTableAssociation	PublicSubnetRouteTableAssociation	
▶ 19:09:22 UTC+0550	CREATE_COMPLETE	AWS::EC2::Route	PublicRoute	
▶ 19:09:22 UTC+0550	CREATE_COMPLETE	AWS::EC2::Subnet	PublicSubnet	
▶ 19:09:19 UTC+0550	CREATE_COMPLETE	AWS::EC2::VPCGatewayAttachment	AttachGateway	

ELB Details

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK &

Create Load Balancer

Actions ▾

Filter by tags and attributes or search by keyword

Name

DNS name

State

VPC ID

Availability Zones

Type

BlueGreenLoadBalancer-5e...

BlueGreenLoadBalancer-5e...

vpc-75a8e21d

ap-south-1a

classic

Load balancer: BlueGreenLoadBalancer-5eeo3gy

Description

Instances

Health Check

Listeners

Monitoring

Tags

Migration

Basic Configuration

Name:

BlueGreenLoadBalancer-5eeo3gy

Creation time:

July 6, 2018 at 7:09:25 PM UTC+5:30

* DNS name:

BlueGreenLoadBalancer-5eeo3gy-1583560865.ap-south-1.elb.amazonaws.com (A Record)

Hosted zone:

ZP97RAFLXTNZK

Type:

Classic (Migrate Now)

Status:

0 of 0 instances in service

Scheme:

internet-facing

VPC:

vpc-75a8e21d

Availability Zones:

subnet-502d5238 - ap-south-1a

Launch Configuration Info

- EC2 Dashboard
- Events
- Tags
- Reports
- Limits
- INSTANCES
 - Instances
 - Launch Templates
 - Spot Requests
 - Reserved Instances
 - Dedicated Hosts
- IMAGES
 - AMIs
 - Bundle Tasks
- ELASTIC BLOCK STORE
 - Volumes
 - Snapshots
- NETWORK & SECURITY
 - Security Groups
 - Elastic IPs
 - Placement Groups
 - Key Pairs

Launch Templates have arrived!

The EC2 Auto Scaling console now has full support for launch templates. Launch templates can be updated and versioned, and include support for the latest features of Amazon EC2. Create an Auto Scaling group to get started or [Learn more](#).

[Create launch configuration](#)[Create Auto Scaling group](#)[Copy to launch template](#)[Actions](#) ▾

Filter:

1 to 1 of 1 Launch Configurations

<input type="checkbox"/>	Name	AMI ID	Instance Type	Spot Price	Creation Time
<input checked="" type="checkbox"/>	CodeDeployB...	ami-34b4c05b	t2.micro		July 6, 2018 at 7:11:12 PM UTC...

Launch Configuration: CodeDeployBGStack-5ee03gy-BlueGreenLaunchConfiguration-20JTJWFU4SG6



Details

[Copy launch configuration](#)

AMI ID	ami-34b4c05b	Instance Type	t2.micro
IAM Instance Profile	CodeDeployBGStack-5ee03gy-InstanceRoleInstanceProfile-T8OSUZQIAI5R	Kernel ID	
Key Name	codeDeploy	Monitoring	true
EBS Optimized	false	Security Groups	sg-0847bf62
Spot Price		Creation Time	Fri Jul 06 19:11:12 GMT+530 2018
RAM Disk ID		Block Devices	-
User data	View User data	IP Address Type	Only assign a public IP address to instances launched in

Auto Scaling Details

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

Key Pairs



Launch Templates have arrived!

The EC2 Auto Scaling console now has full support for launch templates. Launch templates can be updated and versioned, and include support for the latest features of Amazon EC2. Create an Auto Scaling group to get started or [Learn more](#).



Create Auto Scaling group

Actions ▾



Filter:

Filter Auto Scaling groups...



1 to 1 of 1 Auto Scaling Groups

<input type="checkbox"/>	Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
<input checked="" type="checkbox"/>	CodeDeployB...	CodeDeployBGStack-5...	3	3	3	3	ap-south-1a	300	0

Auto Scaling Group: CodeDeployBGStack-5ee03gy-BlueGreenAutoScalingGroup-EYICNSMPZLLT



Details

Activity History

Scaling Policies

Instances

Monitoring

Notifications

Tags

Scheduled Actions

Lifecycle Hooks

Launch Template ⓘ

-

Launch Template Version ⓘ

-

Launch Configuration ⓘ

CodeDeployBGStack-5ee03gy-BlueGreenLaunchConfiguration-20JTJWFU4SG6

Service-Linked Role ⓘ

arn:aws:iam::727203166843:role/aws-service-

Termination Policies ⓘ

Default

Creation Time ⓘ

Fri Jul 06 19:11:16 GMT+530 2018

Availability Zone(s) ⓘ

ap-south-1a

Subnet(s) ⓘ

subnet-502d5238

Default Cooldown ⓘ

300

Placement Groups ⓘ

EC2 Instance Info

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Launch Templates

Spot Requests

Reserved Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

1 to 3 of 3

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
<input checked="" type="checkbox"/>		i-091402141322a7154	t2.micro	ap-south-1a	running	Initializing	None	ec2-35-154-72-108.ap-south-1.compute.amazonaws.com	35.154.72.108
<input type="checkbox"/>		i-0a2fe752c30875460	t2.micro	ap-south-1a	running	Initializing	None	ec2-13-232-24-84.ap-south-1.compute.amazonaws.com	13.232.24.84
<input type="checkbox"/>		i-0abfeb0d94d7d3ef1	t2.micro	ap-south-1a	running	Initializing	None	ec2-13-232-151-178.ap-south-1.compute.amazonaws.com	13.232.151.178

Instance: i-091402141322a7154

Public DNS: ec2-35-154-72-108.ap-south-1.compute.amazonaws.com

Description

Status Checks

Monitoring

Tags

Instance ID	i-091402141322a7154	Public DNS (IPv4)	ec2-35-154-72-108.ap-south-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	35.154.72.108
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-0-19.ap-south-1.compute.internal
Availability zone	ap-south-1a	Private IPs	172.31.0.19

Thank you!