**AWS CodeDeploy**

**Deployment starts,whenever code commits**

AWS CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances or on-premises instances in your own facility.

You can deploy a nearly unlimited variety of application content, such as code, web and configuration files, executables, packages, scripts, multimedia files, and so on. AWS CodeDeploy can deploy application content stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. You do not need to make changes to your existing code before you can use AWS CodeDeploy.

AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications, without many of the risks associated with error-prone manual deployments.

The service scales with your infrastructure so you can easily deploy to one instance or thousands.

AWS CodeDeploy works with various systems for configuration management, source control, continuous integration, continuous delivery, and continuous deployment

## Benefits of AWS CodeDeploy

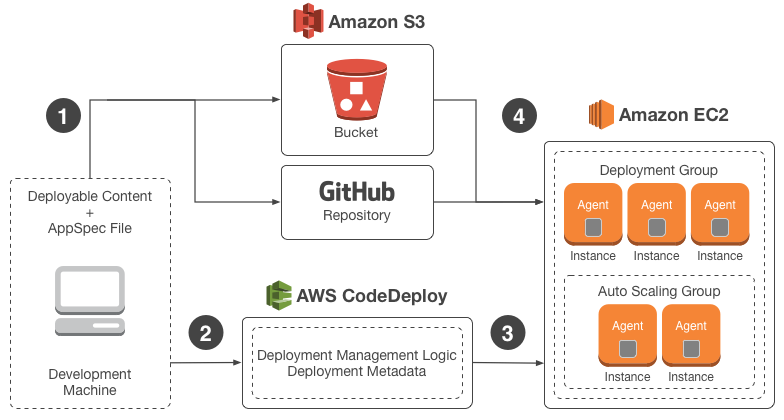
AWS CodeDeploy offers these benefits:

* **Automated deployments**. AWS CodeDeploy fully automates your application deployments across your development, test, and production environments. AWS CodeDeploy scales with your infrastructure so that you can deploy to one instance or thousands.
* **Minimize downtime**. AWS CodeDeploy helps maximize your application availability. During an in-place deployment, AWS CodeDeploy performs a rolling update across Amazon EC2 instances. You can specify the number of instances to be taken offline at a time for updates. During a blue/green deployment, the latest application revision is installed on replacement instances, and traffic is rerouted to these instances when you choose, either immediately or as soon as you are done testing the new environment. For both deployment types, AWS CodeDeploy tracks application health according to rules you configure.
* **Stop and roll back**. You can automatically or manually stop and roll back deployments if there are errors.
* **Centralized control**. You can launch and track the status of your deployments through the AWS CodeDeploy console or the AWS CLI. You will receive a report that lists when each application revision was deployed and to which Amazon EC2 instances.
* **Easy to adopt**. AWS CodeDeploy is platform-agnostic and works with any application. You can easily reuse your setup code. AWS CodeDeploy can also integrate with your software release process or continuous delivery toolchain.

## Overview of AWS CodeDeploy Deployment Types

AWS CodeDeploy provides two deployment type options:

* **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 choose to use a load balancer so each instance is deregistered during its deployment and then restored to service after the deployment is complete.
* **Blue/green deployment**: The instances in a deployment group (the original environment) are replaced by a different set of instances (the replacement environment) using these steps:
  + Instances are provisioned for the replacement environment.
  + The latest application revision is installed on the replacement instances.
  + An optional wait time occurs for activities such as application testing and system verification.
  + Instances in the replacement environment are registered with an Elastic Load Balancing load balancer, causing traffic to be rerouted to them. Instances in the original environment are deregistered and can be terminated or kept running for other uses.



Here's how it works:

1. First, you create deployable content on your local development machine or similar environment, and then you add an application specification file (AppSpec file). The AppSpec file is unique to AWS CodeDeploy; it defines the deployment actions you want AWS CodeDeploy to execute. You bundle your deployable content and the AppSpec file into an archive file, and then upload it to an Amazon S3 bucket or a GitHub repository. This archive file is called an application revision (or simply a revision).
2. Next, you provide AWS CodeDeploy with information about your deployment, such as which Amazon S3 bucket or GitHub repository to pull the revision from and which set of Amazon EC2 instances to deploy its contents to. AWS CodeDeploy calls a set of Amazon EC2 instances a deployment group. A deployment group contains individually tagged Amazon EC2 instances, Amazon EC2 instances in Auto Scaling groups, or both.

Each time you successfully upload a new application revision that you want to deploy to the deployment group, that bundle is set as the target revision for the deployment group. In other words, the application revision that is currently targeted for deployment is the target revision. This is also the revision that will be pulled for automatic deployments.

1. Next, the AWS CodeDeploy agent on each instance polls AWS CodeDeploy to determine what and when to pull from the specified Amazon S3 bucket or GitHub repository.
2. Finally, the AWS CodeDeploy agent on each instance pulls the target revision from the specified Amazon S3 bucket or GitHub repository and, using the instructions in the AppSpec file, deploys the contents to the instance.

AWS CodeDeploy keeps a record of your deployments so that you can get information such as deployment status, deployment configuration parameters, instance health, and so on.

### Overview of a Blue/Green Deployment

A blue/green deployment, in which traffic is rerouted from one set of instances (the original environment) to a different set (the replacement environment), offers a number of advantages over an in-place deployment:

# AWS CodeDeploy Primary Components

Before you start working with the service, you should familiarize yourself with the major components of the AWS CodeDeploy deployment process that are referred to in this user guide.

**Application**: A name that uniquely identifies the application you want to deploy. AWS CodeDeploy uses this name, which functions as a container, to ensure the correct combination of revision, deployment configuration, and deployment group are referenced during a deployment.

**Deployment configuration**: A set of deployment rules and deployment success and failure conditions used by AWS CodeDeploy during a deployment.

**Deployment group**: A set of individual instances. A deployment group contains individually tagged instances, Amazon EC2 instances in Auto Scaling groups, or both**Deployment type**: The method used to make the latest application revision available on instances in a deployment group.

* **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 choose to use a load balancer so each instance is deregistered during its deployment and then restored to service after the deployment is complete.
* **Blue/green deployment**: The instances in a deployment group (the original environment) are replaced by a different set of instances (the replacement environment) using these steps:
  + Instances are provisioned for the replacement environment.
  + The latest application revision is installed on the replacement instances.
  + An optional wait time occurs for activities such as application testing and system verification.
  + Instances in the replacement environment are registered with an Elastic Load Balancing load balancer, causing traffic to be rerouted to them. Instances in the original environment are deregistered and can be terminated or kept running for other uses.

Step 1: Set Up a GitHub Account

Step 2: Create a GitHub Repository

Step 3: Upload a Sample Application to Your GitHub Repository

Step 4: Provision an Instance

Step 5: Create an Application and Deployment Group

Step 6: Deploy the Application to the Instance

Step 7: Monitor and Verify the Deployment

Step 8: Clean Up

http://docs.aws.amazon.com/codedeploy/latest/userguide/tutorials-github.html