What is CodeDeploy?

CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.

You can deploy a nearly unlimited variety of application content, including:

- Code
- Serverless AWS Lambda functions
- Web and configuration files
- Executables
- Packages
- Scripts
- Multimedia files

CodeDeploy can deploy application content that runs on a server and is stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. CodeDeploy can also deploy a serverless Lambda function. You do not need to make changes to your existing code before you can use CodeDeploy.

CodeDeploy makes it easier for you to:

- Rapidly release new features.
- Update AWS Lambda function versions.
- Avoid downtime during application deployment.
- Handle 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.

CodeDeploy works with various systems for configuration management, source control, continuous integration, continuous delivery, and continuous deployment. For more information, see Product integrations.

The CodeDeploy console also provides a way to quickly search for your resources, such as repositories, build projects, deployment applications, and pipelines. Choose **Go to resource** or press the / key, and then type the name of the resource. Any matches appear in the list. Searches are case insensitive. You only see resources that you have permissions to view. For more information, see Identity and access management for AWS CodeDeploy (p. 357).

Topics

Benefits of AWS CodeDeploy

CodeDeploy offers these benefits:

- **Server, serverless, and container applications**. CodeDeploy lets you deploy both traditional applications on servers and applications that deploy a serverless AWS Lambda function version or an Amazon ECS application.
- **Automated deployments**. CodeDeploy fully automates your application deployments across your development, test, and production environments. CodeDeploy scales with your infrastructure so that you can deploy to one instance or thousands.
- **Minimize downtime**. If your application uses the EC2/On-Premises compute platform, CodeDeploy helps maximize your application availability. During an in-place deployment, 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. 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, 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 CodeDeploy console or the AWS CLI. You receive a report that lists when each application revision was deployed and to which Amazon EC2 instances.
- **Easy to adopt**. CodeDeploy is platform-agnostic and works with any application. You can easily reuse your setup code. CodeDeploy can also integrate with your software release process or continuous delivery toolchain.
- **Concurrent deployments**. If you have more than one application that uses the EC2/On-Premises compute platform, CodeDeploy can deploy them concurrently to the same set of instances.

Overview of CodeDeploy compute platforms

CodeDeploy is able to deploy applications to three compute platforms:

- **EC2/On-Premises**: Describes instances of physical servers that can be Amazon EC2 cloud instances, on-premises servers, or both. Applications created using the EC2/On-Premises compute platform can be composed of executable files, configuration files, images, and more.

  Deployments that use the EC2/On-Premises compute platform manage the way in which traffic is directed to instances by using an in-place or blue/green deployment type. For more information, see Overview of CodeDeploy deployment types (p. 6).
- **AWS Lambda**: Used to deploy applications that consist of an updated version of a Lambda function. AWS Lambda manages the Lambda function in a serverless compute environment made up of a high-availability compute structure. All administration of the compute resources is performed by AWS Lambda. For more information, see Serverless Computing and Applications. For more information about AWS Lambda and Lambda functions, see AWS Lambda.

  You can manage the way in which traffic is shifted to the updated Lambda function versions during a deployment by choosing a canary, linear, or all-at-once configuration.
- **Amazon ECS**: Used to deploy an Amazon ECS containerized application as a task set. CodeDeploy performs a blue/green deployment by installing an updated version of the application as a new replacement task set. CodeDeploy reroutes production traffic from the original application task set to

the replacement task set. The original task set is terminated after a successful deployment. For more information about Amazon ECS, see Amazon Elastic Container Service.

You can manage the way in which traffic is shifted to the updated task set during a deployment by choosing a canary, linear, or all-at-once configuration.

Note
Amazon ECS blue/green deployments are supported using both CodeDeploy and AWS CloudFormation. Details for these deployments are described in subsequent sections.

The following table describes how CodeDeploy components are used with each compute platform. For more information, see:

- Working with deployment groups in CodeDeploy (p. 272)
- Working with deployments in CodeDeploy (p. 306)
- Working with deployment configurations in CodeDeploy (p. 251)
- Working with application revisions for CodeDeploy (p. 291)
- Working with applications in CodeDeploy (p. 259)

CodeDeploy component	EC2/On-Premises	AWS Lambda	Amazon ECS
Deployment group	Deploys a revision to a set of instances.	Deploys a new version of a serverless Lambda function on a high-availability compute infrastructure.	Specifies the Amazon ECS service with the containerized application to deploy as a task set, a production and optional test listener used to serve traffic to the deployed application, when to reroute traffic and terminate the deployed application's original task set, and optional trigger, alarm, and rollback settings.
Deployment	Deploys a new revision that consists of an application and AppSpec file. The	Shifts production traffic from one version of a Lambda function to a	Deploys an updated version of an

CodeDeploy component	EC2/On-Premises	AWS Lambda	Amazon ECS
	AppSpec specifies how to deploy the application to the instances in a deployment group.	new version of the same function. The AppSpec file specifies which Lambda function version to deploy.	Amazon ECS containerized application as a new, replacement task set. CodeDeploy reroutes production traffic from the task set with the original version to the new replacement task set with the updated version. When the deployment completes, the original task set is terminated.
Deployment configuration	Settings that determine the deployment speed and the minimum number of instances that must be healthy at any point during a deployment.	Settings that determine how traffic is shifted to the updated Lambda function versions.	Settings that determine how traffic is shifted to the updated Amazon ECS task set.

CodeDeploy component	EC2/On-Premises	AWS Lambda	Amazon ECS
Revision	A combination of an AppSpec file and application files, such as executables, configuration files, and so on.	An AppSpec file that specifies which Lambda function to deploy and Lambda functions that can run validation tests during deployment lifecycle event hooks.	An AppSpec file that specifies: • The Amazon ECS task definition for the Amazon ECS service with the containerized application to deploy. • The container where your updated application is deployed. • A port for the container where production traffic is rerouted. • Optional network configuration settings and Lambda functions that can run validation tests during deployment lifecycle event hooks.

CodeDeploy component	EC2/On-Premises	AWS Lambda	Amazon ECS
Application	A collection of deployment groups and revisions. An EC2/On-Premises application uses the EC2/On-Premises compute platform.	A collection of deployment groups and revisions. An application used for an AWS Lambda deployment uses the serverless AWS Lambda compute platform.	A collection of deployment groups and revisions. An application used for an Amazon ECS deployment uses the Amazon ECS compute platform.

Overview of CodeDeploy deployment types

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 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. For more information about in-place deployments, see Overview of an in-place deployment (p. 7).

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

- **Blue/green deployment**: The behavior of your deployment depends on which compute platform you use:
  - **Blue/green on an EC2/On-Premises compute platform**: 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 one or more Elastic Load Balancing load balancers, causing traffic to be rerouted to them. Instances in the original environment are deregistered and can be terminated or kept running for other uses.

      Note
      If you use an EC2/On-Premises compute platform, be aware that blue/green deployments work with Amazon EC2 instances only.
  - **Blue/green on an AWS Lambda or Amazon ECS compute platform**: Traffic is shifted in increments according to a **canary**, **linear**, or **all-at-once** deployment configuration.
  - **Blue/green deployments through AWS CloudFormation**: Traffic is shifted from your current resources to your updated resources as part of an AWS CloudFormation stack update. Currently, only ECS blue/green deployments are supported.

  For more information about blue/green deployments, see Overview of a blue/green deployment (p. 7).