Deploy Code to a Virtual Machine

with AWS CodeDeploy

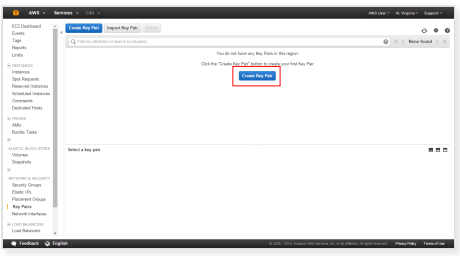
In this tutorial, you will learn how to deploy application code to a virtual machine on AWS. You will use AWS CodeDeploy, a service that automates code deployments to AWS or on-premises servers, to deploy code to virtual machines that you create and manage with Amazon EC2.

Everything done in this tutorial is free tier eligible.

Step 1: Create a Key Pair

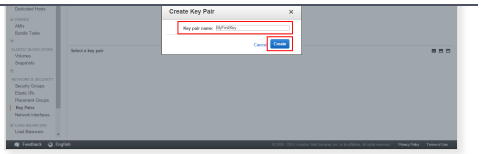
You will need to create a key pair to access your virtual machine with Amazon EC2. If you already have a key pair, skip ahead to Step 2.

a. When you click here, the AWS Management Console will open in a new browser window, so you can keep this step-by-step guide open. Click **Create Key Pair**.



(click to zoom)

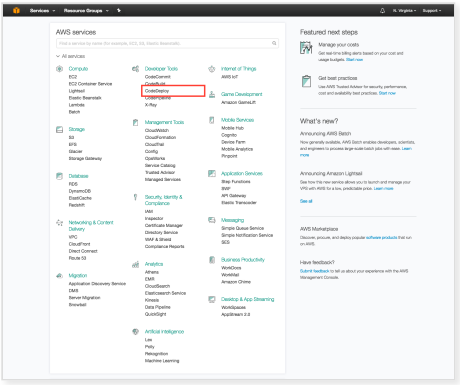
Note: Amazon EC2 uses public-key cryptography to encrypt and decrypt login information. To learn more about key pairs, see Amazon EC2 Key Pairs.



(click to zoom)

Step 2: Enter the CodeDeploy Console

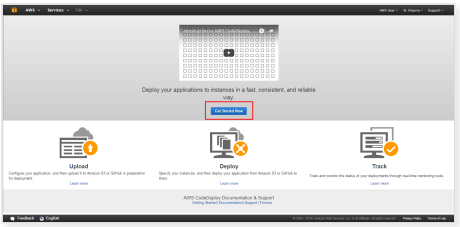
a. Click the home icon on the upper left corner of the AWS Management Console. Find **CodeDeploy** under *Developer Tools* and click to open the AWS CodeDeploy Console.



(click to zoom)

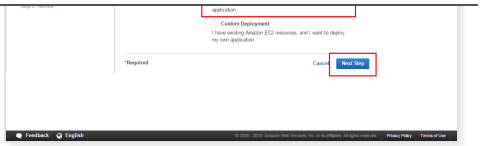
b. In the AWS CodeDeploy Console, click **Get Started Now.**

If you already have applications, look to the right column and click **Create Deployment Walkthrough.**



(click to zoom)

c. Select *Sample Deployment* and click **Next Step**.

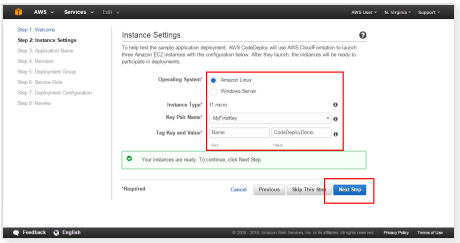


(click to zoom)

Step 3: Launch a Virtual Machine

You will need to launch an AWS virtual machine to deploy your code on. AWS virtual machines are known as Amazon EC2 instances, or just 'instances' for short. In this step, we will launch three EC2 instances using a pre-configured EC2 template.

You will configure your instance settings with the options below:



(click to zoom)

- **Operating System**: You can choose the OS of your EC2 Instance. For this tutorial, we will use *Amazon Linux*.
- **Instance Type**: For this tutorial, the *t1.micro* instance type has been selected as the default value to stay within the free tier. Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications.
- **Key Pair Name**: From the drop-down list, choose the Amazon EC2 instance key pair you created in step 1, *MyFirstKey*, to connect to the Amazon EC2 instances. You can also choose a key pair you already have.
- **Tag Key and Value:** AWS CodeDeploy will use this tag key and value to locate the instances during deployments. You can leave the default values.

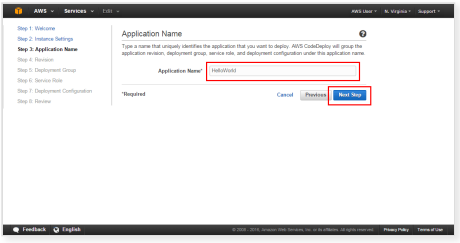
Note: This step may take several minutes to complete. As you wait, feel free to review this tutorial with the video below:

AWS CodeDeploy Walkthrough

Step 4: Name Your Application and Review Your Application Revision

AWS CodeDeploy uses application names during code deployments to make sure it is referencing the correct deployment components, such as the deployment group, deployment configuration, and application revision.

a. In the *Application Name* box, enter *HelloWorld* as the name for your sample application and click **Next Step.**



(click to zoom)

(click to zoom)

Note: You have the option to download the sample bundle. In this view, you can review information about the application revision you'd like to deploy to EC2. An application revision is an archive file containing source content—such as source code, web pages, executable files, and deployment scripts—along with an application specification file (AppSpec file). The AppSpec file helps CodeDeploy map the source files in your revision to their destinations and run scripts at various stages of the deployment.

Click **Next Step**

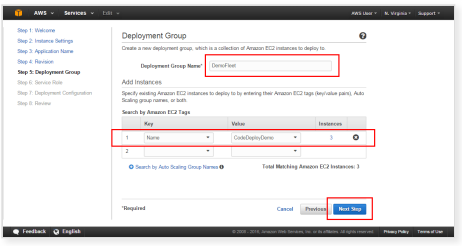
Step 5: Create a Deployment Group

A deployment group is a set of individual EC2 instances that CodeDeploy deploys revisions to. A deployment group contains individually tagged instances, Amazon EC2 instances in Auto Scaling groups, or both.

In the Deployment Group Name box, leave the proposed deployment group name (*DemoFleet*) as is.

You will then specify the Amazon EC2 instances to deploy by entering the key-value pair in the *Seach by Amazon EC2 Tags* section:

- The *Key* and *Value* columns should be autopopulated with the values from Step 3.



(click to zoom)

we have launched and pre-configured three EC2 instances and these instances have already been tagged together into a deployment group.

Choose **Next Step.**

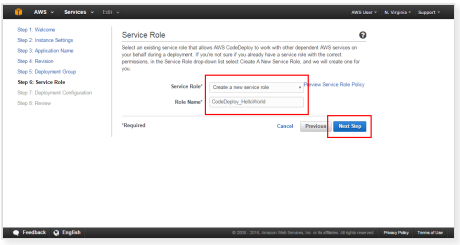
Step 6: Create a Service Role

In this step, you will grant AWS CodeDeploy permission to deploy to your instances. You create a role for an AWS service when you want to grant permissions to a service like Amazon EC2 or AWS CodeDeploy. These services can access AWS resources, so you create a role to determine what the service is allowed to do with those resources.

Service Role: Choose *Create a new service role*. If you already have a service role, you can choose *Use an existing service role*.

Role Name: You can accept default value of *CodeDeploy_HelloWorld*. If you are using an existing service role, choose it from the Role Name drop-down list.

Click **Next Step**.



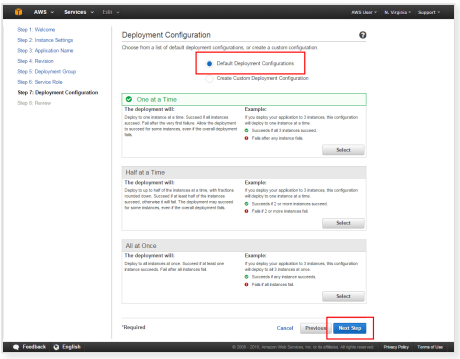
(click to zoom)

Step 7: Deploy Your Application

In this step, we will select a deployment configuration and then initiate the deployment to our three EC2 instances. By the end of this step, we'll have successfully deployed a live and

a. The deployment configuration lets you determine how many instances to simultaneously deploy your application revisions to and describes the success and failure conditions for the deployment. For example, using the default configuration ("One at a Time"), if you deploy your application to 3 instances, this configuration will deploy to one instance at a time.

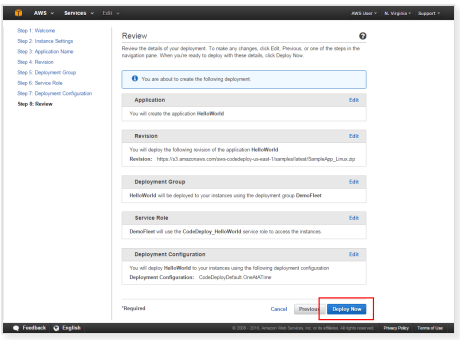
Accept the *Default Deployment Configuration* and click **Next Step.**



(click to zoom)

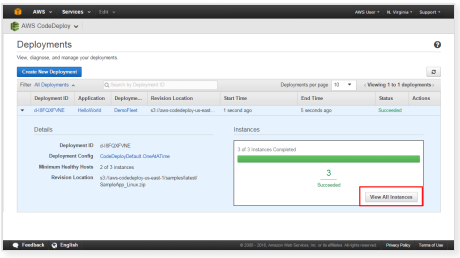
b. Review the details of your deployment and click **Deploy Now.**

Note: This can take several minutes to complete.



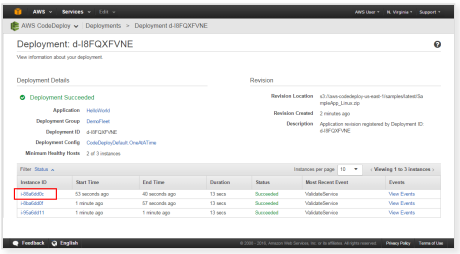
(click to zoom)

c. Our sample revision deploys a single web page to each instance. Once all three instances are completed, click **View All Instances.**



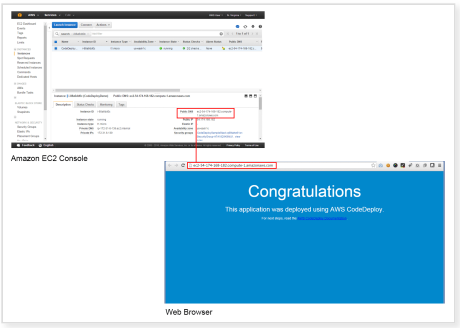
(click to zoom)

d. Click the instance ID for one of the instances you deployed to. This will take you to the EC2 dashboard where you can view the instance that you launched.



(click to zoom)

e. To verify whether your sample application deployed successfully, copy the address in the Public DNS field in the bottom panel, paste the address into your browser, and you will see your live web page.

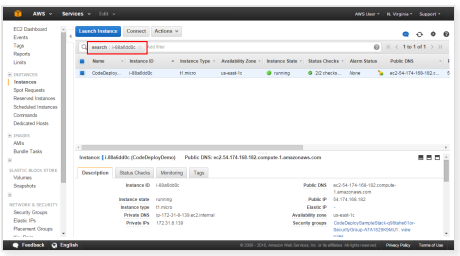


(click to zoom)

Step 8: Clean Up Your Instances

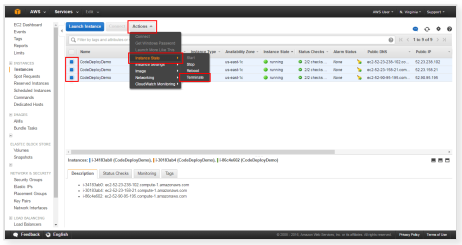
To avoid future charges, you must clean up the resources used in this tutorial. The EC2 instances you launched for this tutorial will keep running unless you terminate them.

a. In the EC2 console, the search bar is autopopulated with a search filter for the Instance ID. Delete this filter and you will see all the instances launched by CodeDeploy.



b. Select the boxes of each Amazon EC2 instance to terminate. Select Actions, Instance State, and click **Terminate**.

When prompted, click **Yes, Terminate**.



(click to zoom)

Congratulations!

You have successfully created and completed your very first code deployment to Amazon EC2 instances using AWS CodeDeploy. Using a provided template, you first launched three instances that were pre-configured with the proper tags and installed with the agent necessary to use AWS CodeDeploy. Finally, you prepared your application for deployment, granted CodeDeploy permission to deploy to your instances, and then successfully deployed your code.

Next Steps

Now that you have completed a sample deployment to Amazon EC2 instances using AWS CodeDeploy, you can begin learning how to use CodeDeploy in your own applications. Visit the CodeDeploy documentation to learn how to launch EC2 instances and then configure the instances so that CodeDeploy can deploy code to them. In our tutorial, we simplified those configuration steps to help you learn and experience how CodeDeploy works.

Learn how to configure AWS CodeDeploy instances

Was this page helpful?

Yes No

Sign In to the Console

Learn About AWS

What Is AWS?
What Is Cloud Computing?
What Is DevOps?
What Is a Container?
What Is a Data Lake?
AWS Cloud Security
What's New
Blogs
Press Releases

Resources for AWS

Getting Started
Training and Certification
AWS Solutions Portfolio
Architecture Center
Product and Technical FAQs
Analyst Reports
AWS Partner Network

Developers on AWS

Developer Center
SDKs & Tools
Python on AWS
Java on AWS
PHP on AWS
Javascript on AWS

Help

Contact Us
AWS Careers
File a Support Ticket
Knowledge Center
AWS Support Overview
Legal

Create an AWS Account

Language

عربي |
Bahasa Indonesia |
Deutsch |
English |
Español |
Français |
Italiano |
Português |
Tiếng Việt |
Türkçe |
Русский |
ไทย |
日本語 |
한국어 |
中文 (简体) |
中文 (繁體)