Experiment No. : 09

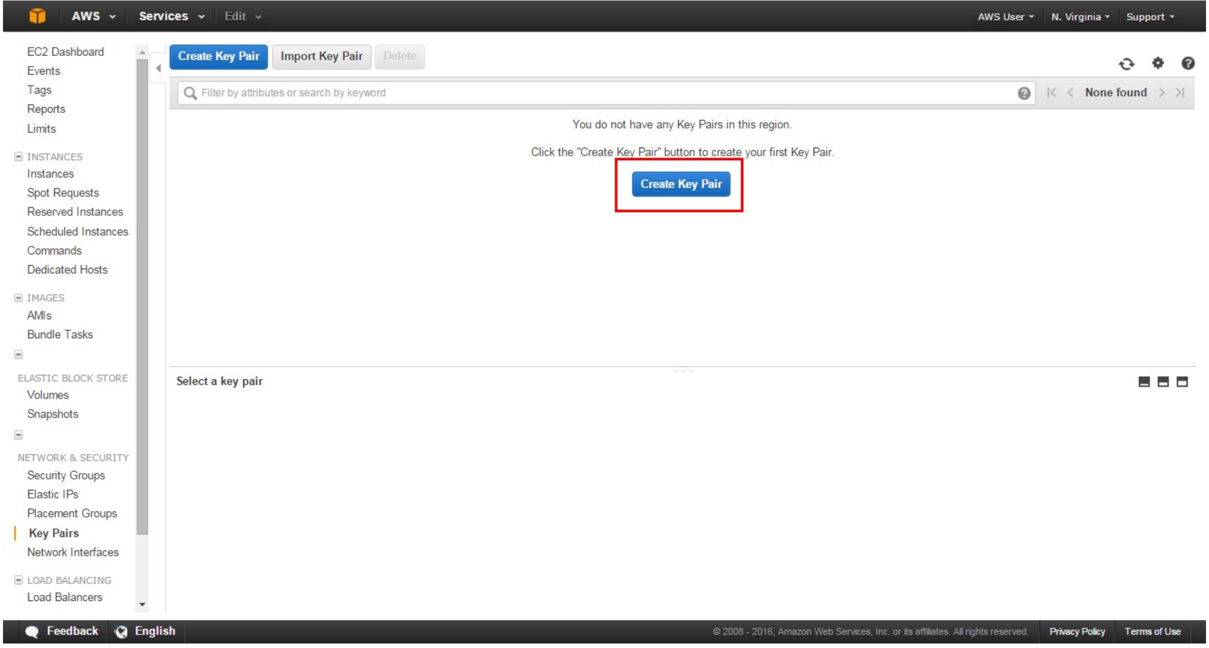
Title: Deploy Code to a Virtual Machine:Learn how to easily deploy code to virtual machines in the cloud.

In this experiment, 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.

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.

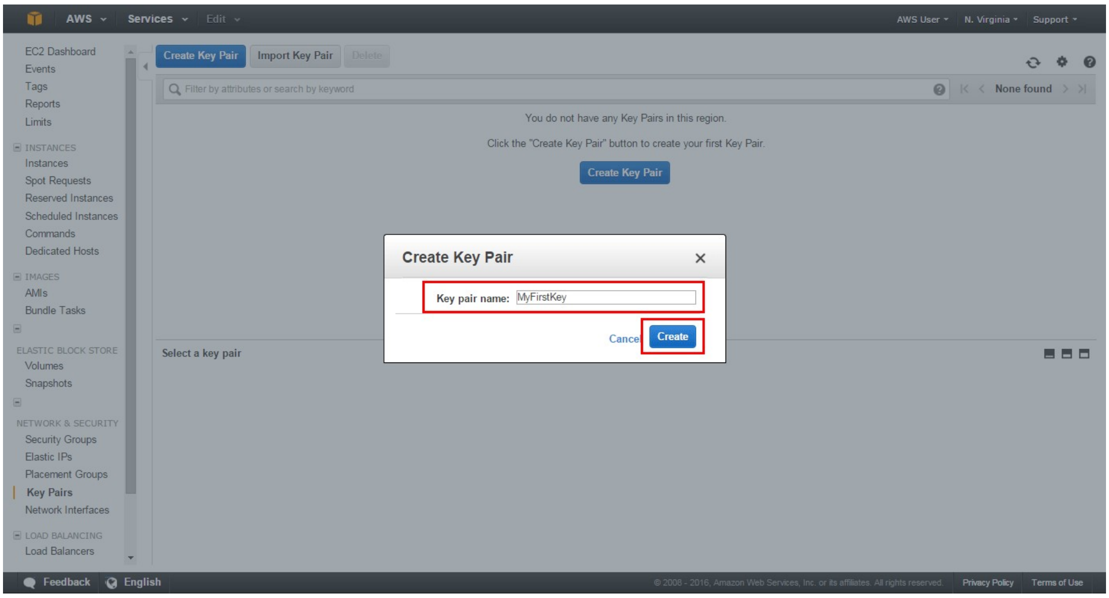


You do not have any Key Pairs in this region.

Click the "Create Key Pair" button to create your first Key Pair.

b. Name your key pair. For this tutorial, we will use MyFirstKey. Click Create.

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



You do not have any Key Pairs in this region.

Click the "Create Key Pair" button to create your first Key Pair.

Create Key Pair

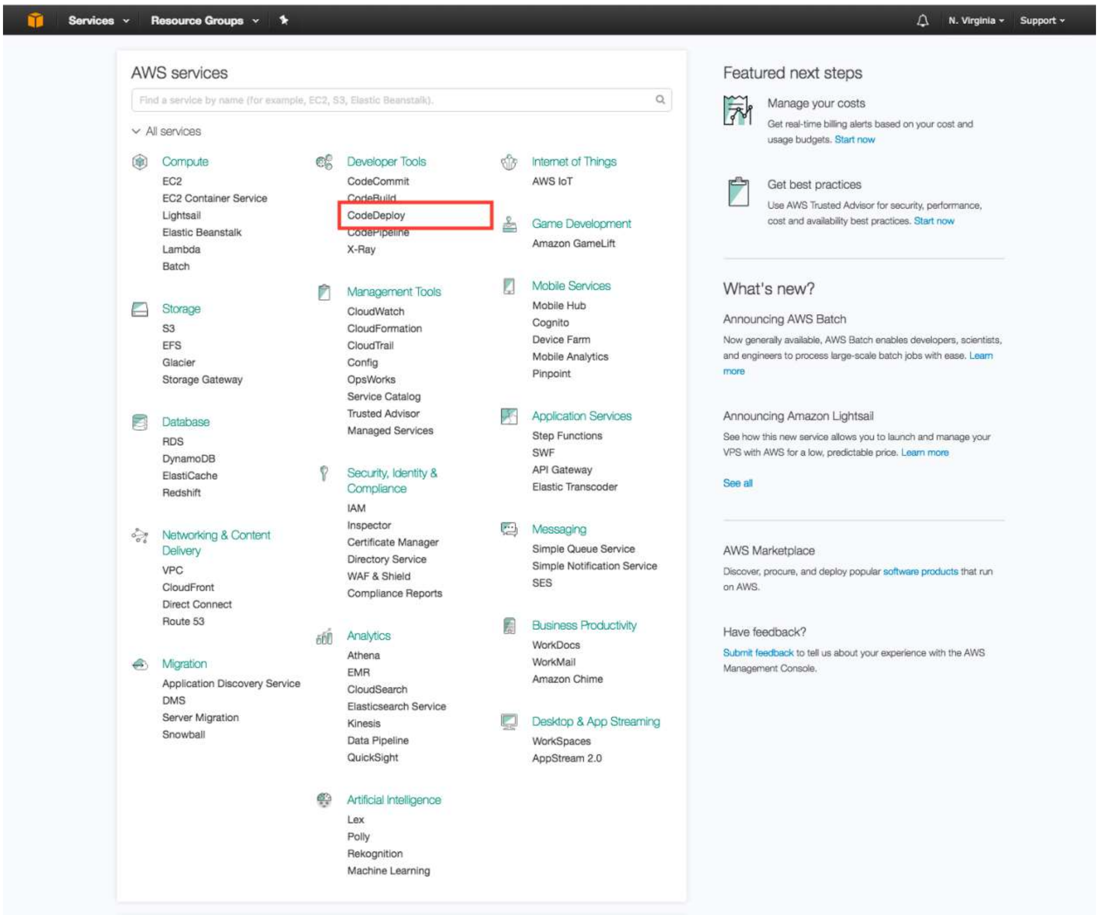
Key pair name: MyFirstKey

Cancel Create

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.



Featured next steps

Manage your costs
Get real-time billing alerts based on your cost and usage budgets. Start now

Get best practices
Use AWS Trusted Advisor for security, performance, cost and availability best practices. Start now

What's new?

Announcing AWS Batch
Now generally available, AWS Batch enables developers, scientists, and engineers to process large-scale batch jobs with ease. Learn more

Announcing Amazon Lightsail
See how this new service allows you to launch and manage your VPS with AWS for a low, predictable price. Learn more

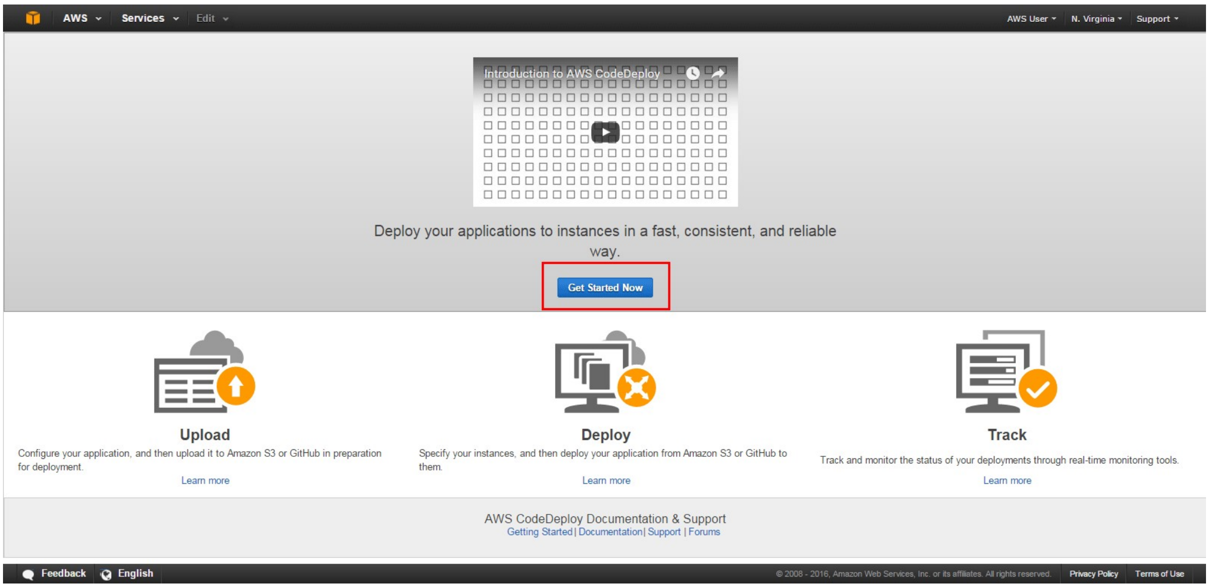
See all

AWS Marketplace
Discover, procure, and deploy popular software products that run on AWS.

Have feedback?
Submit feedback to tell us about your experience with the AWS Management Console.

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.



Deploy your applications to instances in a fast, consistent, and reliable way.

Configure your application, and then upload it to Amazon S3 or GitHub in preparation for deployment.

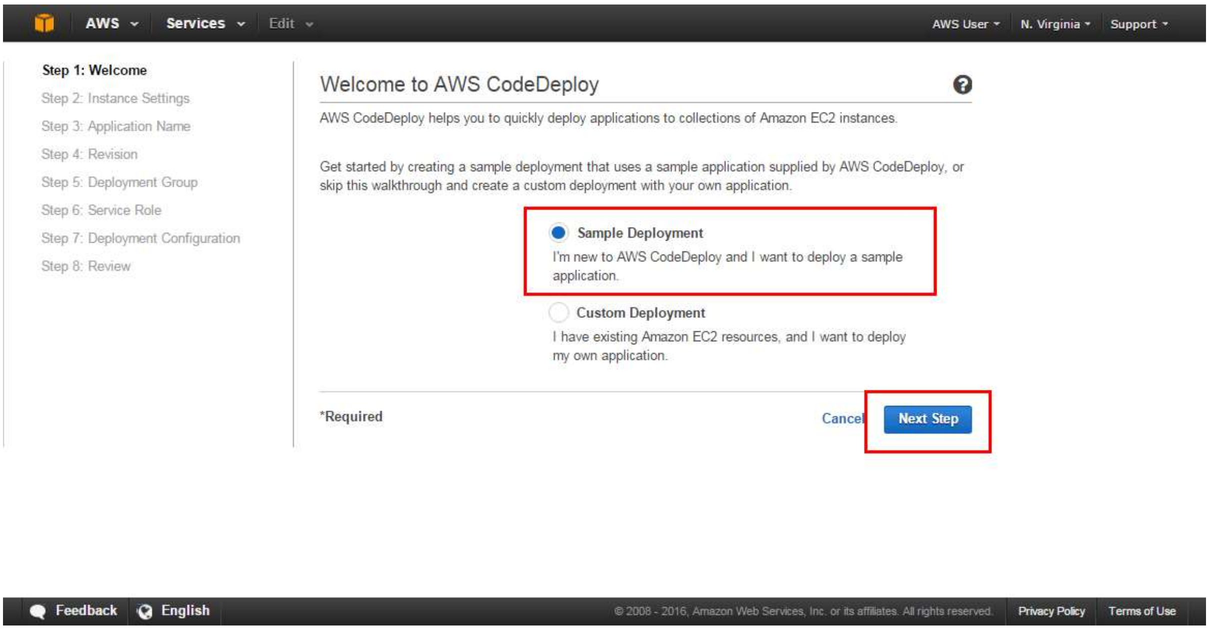
Specify your instances, and then deploy your application from Amazon S3 or GitHub to them.

Track and monitor the status of your deployments through real-time monitoring tools.

AWS CodeDeploy Documentation & Support
Getting Started | Documentation | Support | Forums

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

c. Select Sample Deployment and click Next Step.



Welcome to AWS CodeDeploy

AWS CodeDeploy helps you to quickly deploy applications to collections of Amazon EC2 instances.

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

● Sample Deployment
I'm new to AWS CodeDeploy and I want to deploy a sample application.

○ Custom Deployment
I have existing Amazon EC2 resources, and I want to deploy my own application.

*Required

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Step 3: Launch a Virtual Machine

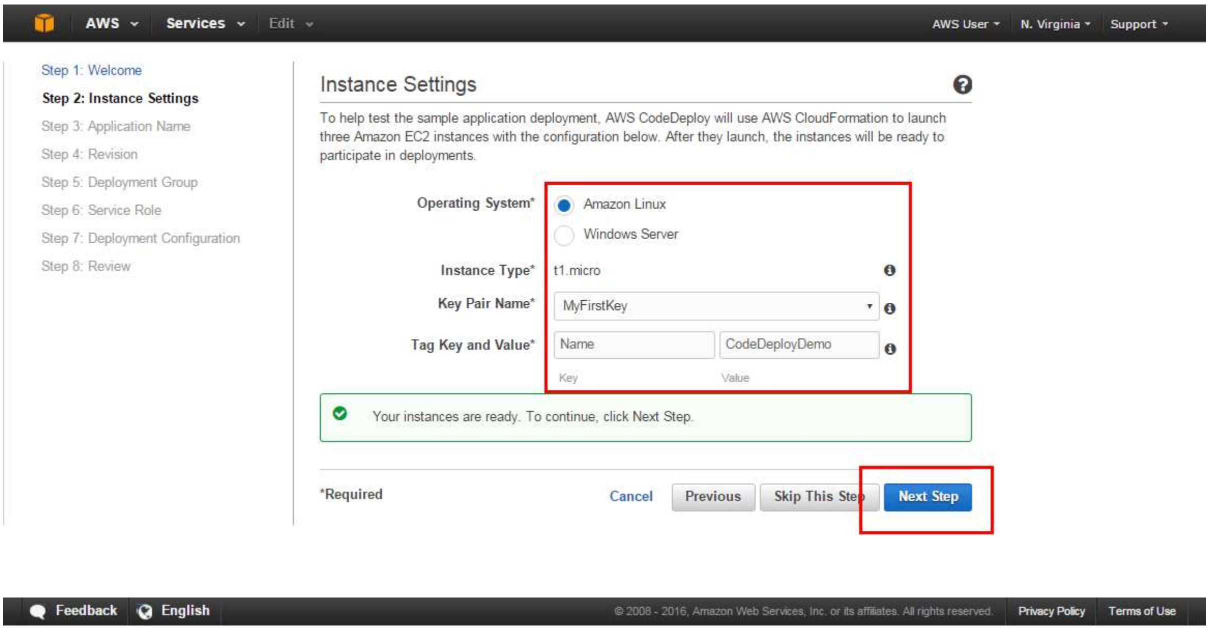
You will need to launch an AWS virtual machine to deploy your code on. AWS virtual machines are known as Amazon EC2 instances, or just 'instances' for short. In this step, we will launch three EC2 instances using a pre-configured EC2 template.

You will configure your instance settings with the options below:

- 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.

Click Launch Instances.

Note: This step may take several minutes to complete.



Instance Settings

To help test the sample application deployment, AWS CodeDeploy will use AWS CloudFormation to launch three Amazon EC2 instances with the configuration below. After they launch, the instances will be ready to participate in deployments.

Operating System* ● Amazon Linux ○ Windows Server

Instance Type* t1.micro

Key Pair Name* MyFirstKey

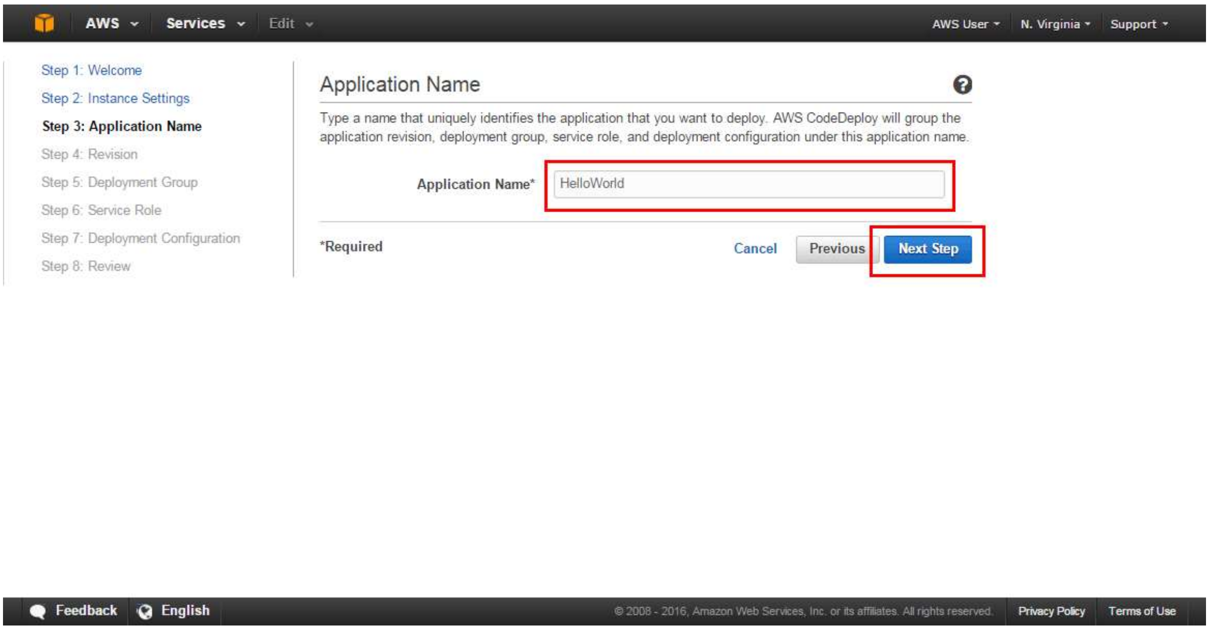
Tag Key and Value* Name (Key) CodeDeployDemo (Value)

Your instances are ready. To continue, click Next Step.

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.



Application Name

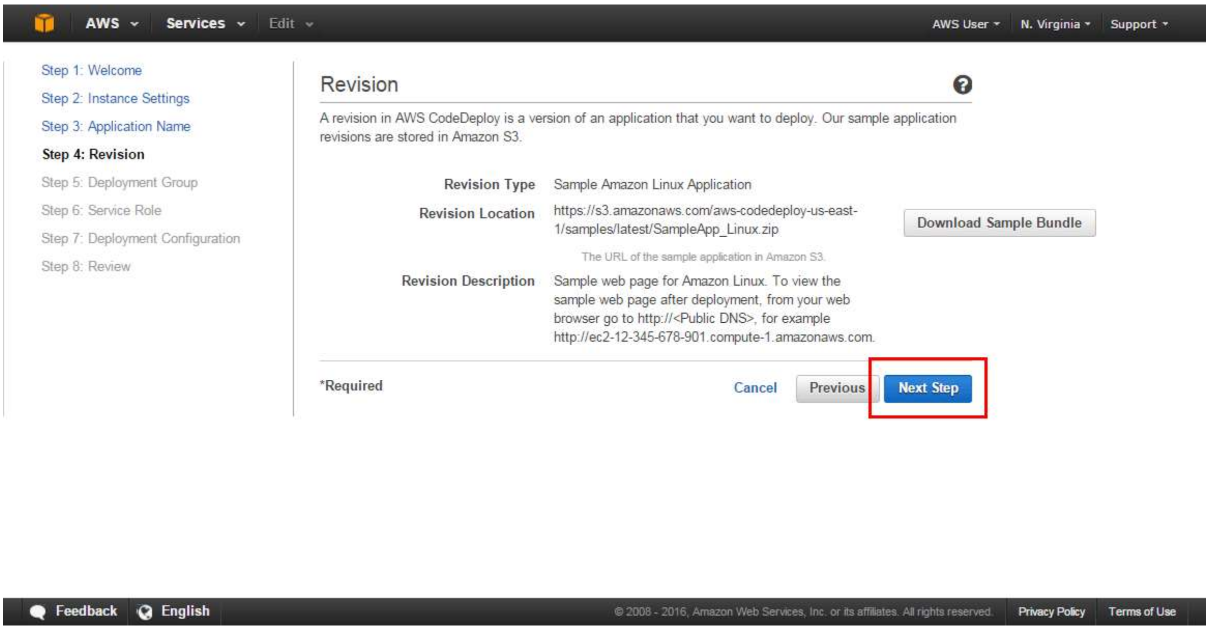
Type a name that uniquely identifies the application that you want to deploy. AWS CodeDeploy will group the application revision, deployment group, service role, and deployment configuration under this application name.

Application Name* HelloWorld

*Required

b. Review information about your application revision, such as its location and description.

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



Revision

A revision in AWS CodeDeploy is a version of an application that you want to deploy. Our sample application revisions are stored in Amazon S3.

Revision Type: Sample Amazon Linux Application

Revision Location: https://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip

The URL of the sample application in Amazon S3.

Revision Description: Sample web page for Amazon Linux. To view the sample web page after deployment, from your web browser go to http://<Public DNS>, for example http://ec2-12-345-678-901.compute-1.amazonaws.com.

*Required

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.

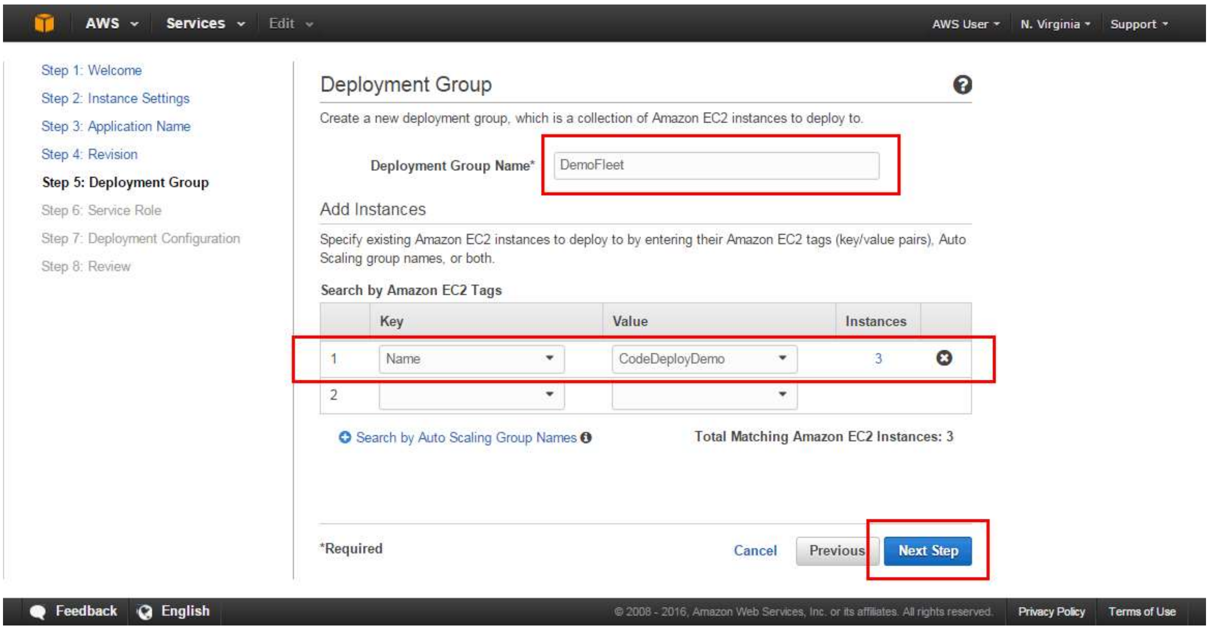
In the Deployment Group Name box, leave the proposed deployment group name (DemoFleet) as is.

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.

The Instances column displays the number of EC2 instances that you will deploy code to. For this tutorial, we have launched and pre-configured three EC2 instances and these instances have already been tagged together into a deployment group.

Choose Next Step.



Deployment Group

Create a new deployment group, which is a collection of Amazon EC2 instances to deploy to.

Deployment Group Name* DemoFleet

Add Instances

Specify existing Amazon EC2 instances to deploy to by entering their Amazon EC2 tags (key/value pairs), Auto Scaling group names, or both.

Search by Amazon EC2 Tags

	Key	Value	Instances
1	Name	CodeDeployDemo	3
2			

Total Matching Amazon EC2 Instances: 3

*Required

Search by Auto Scaling Group Names
Cancel
Previous
Next Step
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

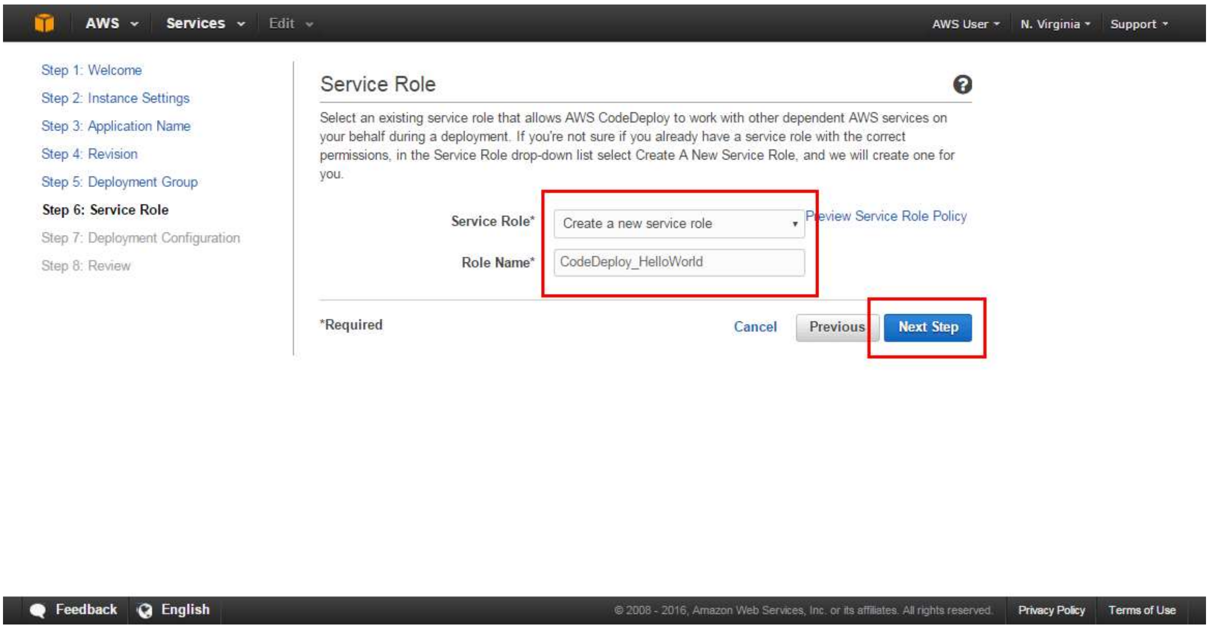
Step 6: Create a Service Role

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.



Service Role

Select an existing service role that allows AWS CodeDeploy to work with other dependent AWS services on your behalf during a deployment. If you're not sure if you already have a service role with the correct permissions, in the Service Role drop-down list select Create A New Service Role, and we will create one for you.

Service Role*: Create a new service role

Role Name*: CodeDeploy_HelloWorld

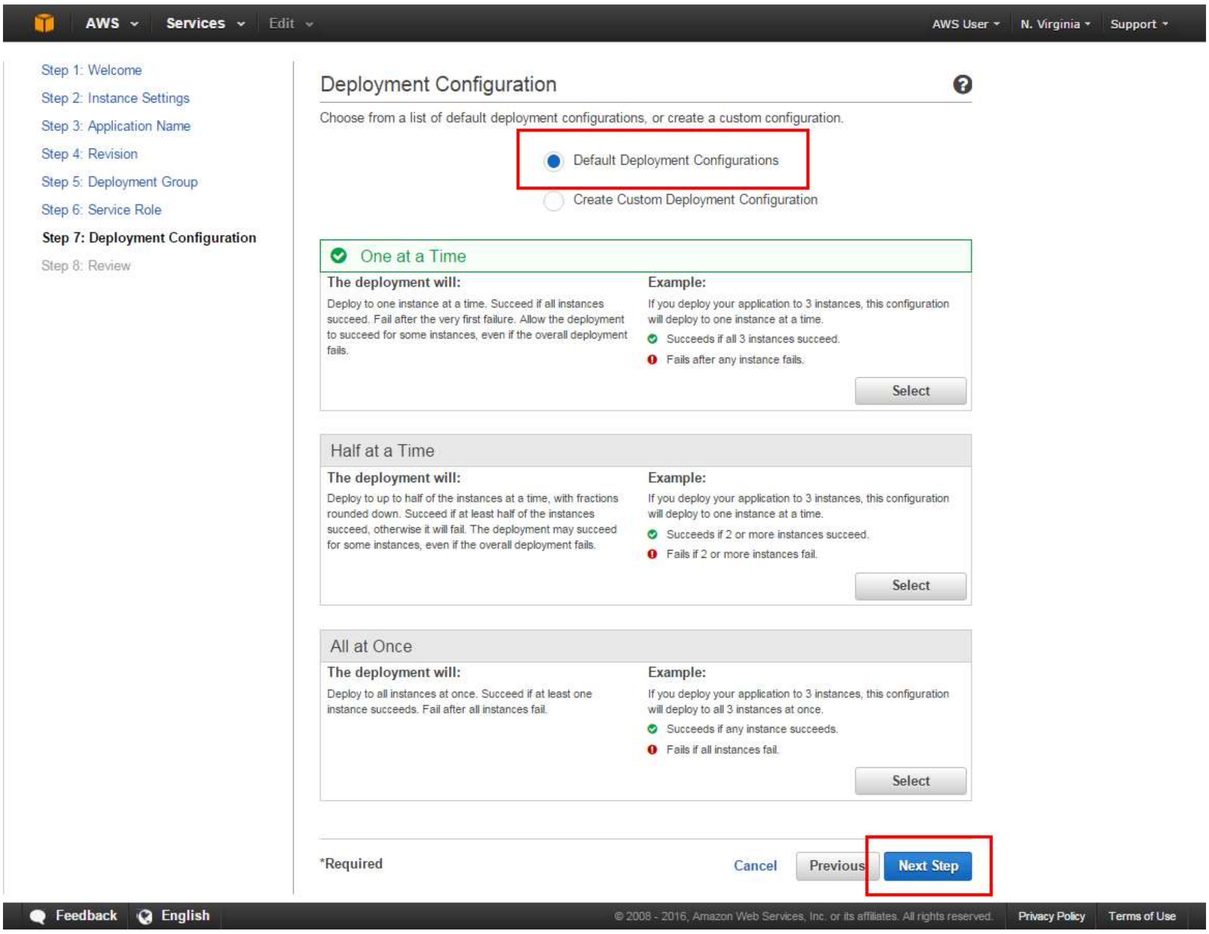
*Required

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 running website, which we can visit online.

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.



Deployment Configuration

Choose from a list of default deployment configurations, or create a custom configuration.

● Default Deployment Configurations

○ Create Custom Deployment Configuration

One at a Time

The deployment will:
Deploy to one instance at a time. Succeed if all instances succeed. Fail after the very first failure. Allow the deployment to succeed for some instances, even if the overall deployment fails.

Example:
If you deploy your application to 3 instances, this configuration will deploy to one instance at a time.
- Succeeds if all 3 instances succeed.
- Fails after any instance fails.

Select

Half at a Time

The deployment will:
Deploy to up to half of the instances at a time, with fractions rounded down. Succeed if at least half of the instances succeed, otherwise it will fail. The deployment may succeed for some instances, even if the overall deployment fails.

Example:
If you deploy your application to 3 instances, this configuration will deploy to one instance at a time.
- Succeeds if 2 or more instances succeed.
- Fails if 2 or more instances fail.

Select

All at Once

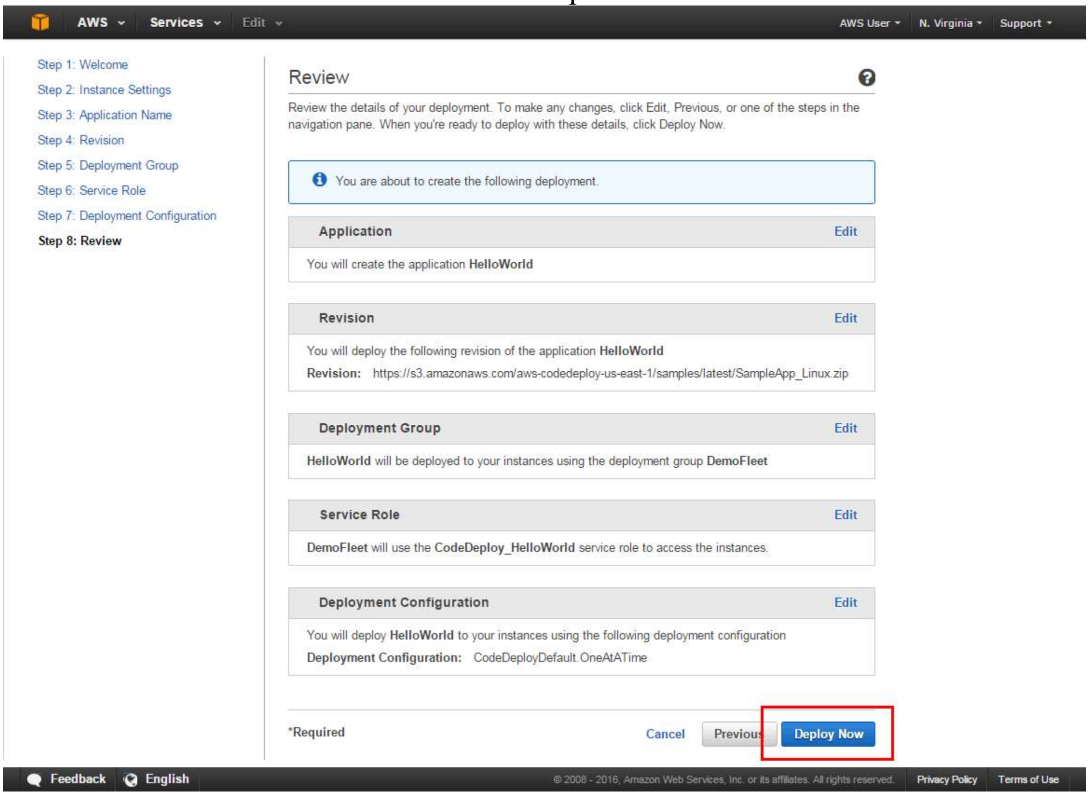
The deployment will:
Deploy to all instances at once. Succeed if at least one instance succeeds. Fail after all instances fail.

Example:
If you deploy your application to 3 instances, this configuration will deploy to all 3 instances at once.
- Succeeds if any instance succeeds.
- Fails if all instances fail.

Select

*Required Cancel Previous Next Step

b. Review the details of your deployment and click Deploy Now.
Note: This can take several minutes to complete.



Review

Review the details of your deployment. To make any changes, click Edit, Previous, or one of the steps in the navigation pane. When you're ready to deploy with these details, click Deploy Now.

You are about to create the following deployment.

Application Edit
You will create the application **HelloWorld**

Revision Edit
You will deploy the following revision of the application **HelloWorld**
Revision: https://s3.amazonaws.com/aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip

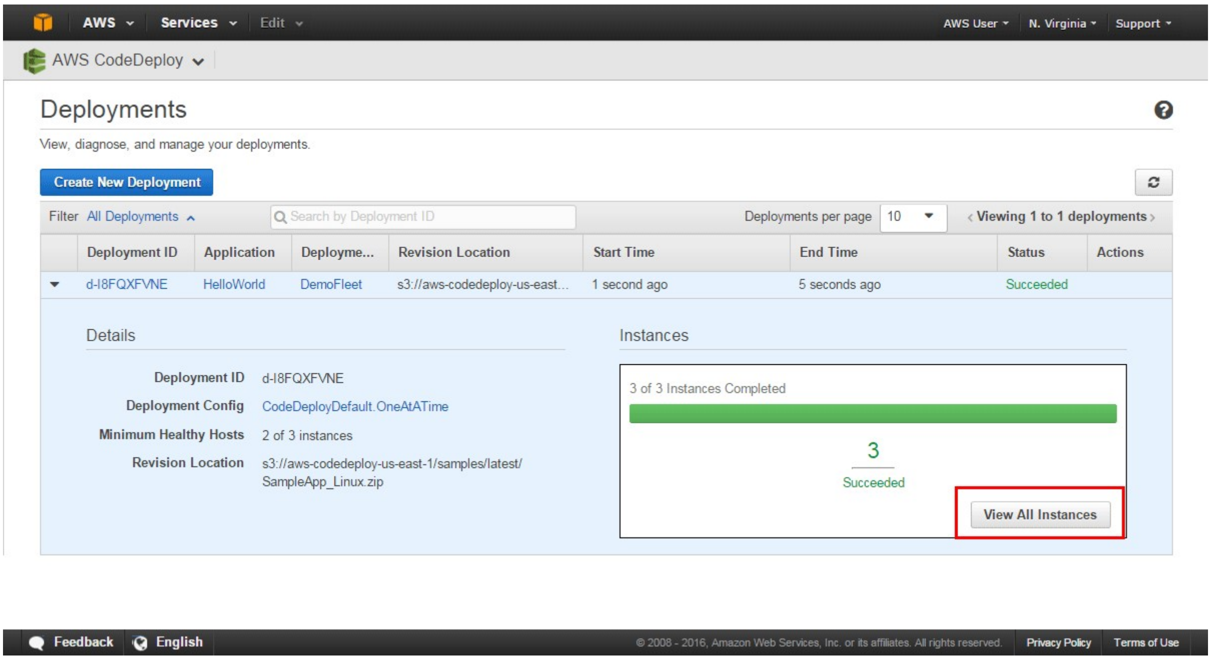
Deployment Group Edit
HelloWorld will be deployed to your instances using the deployment group **DemoFleet**

Service Role Edit
DemoFleet will use the **CodeDeploy_HelloWorld** service role to access the instances.

Deployment Configuration Edit
You will deploy **HelloWorld** to your instances using the following deployment configuration
Deployment Configuration: CodeDeployDefault.OneAtATime

*Required Cancel Previous Deploy Now

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



Deployments

View, diagnose, and manage your deployments.

Filter All Deployments | Search by Deployment ID | Deployments per page 10 | Viewing 1 to 1 deployments

Deployment ID	Application	Deployme...	Revision Location	Start Time	End Time	Status	Actions
d-I8FQXFVNE	HelloWorld	DemoFleet	s3://aws-codedeploy-us-east...	1 second ago	5 seconds ago	Succeeded	

Details

Deployment ID: d-I8FQXFVNE
Deployment Config: CodeDeployDefault.OneAtATime
Minimum Healthy Hosts: 2 of 3 instances
Revision Location: s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip

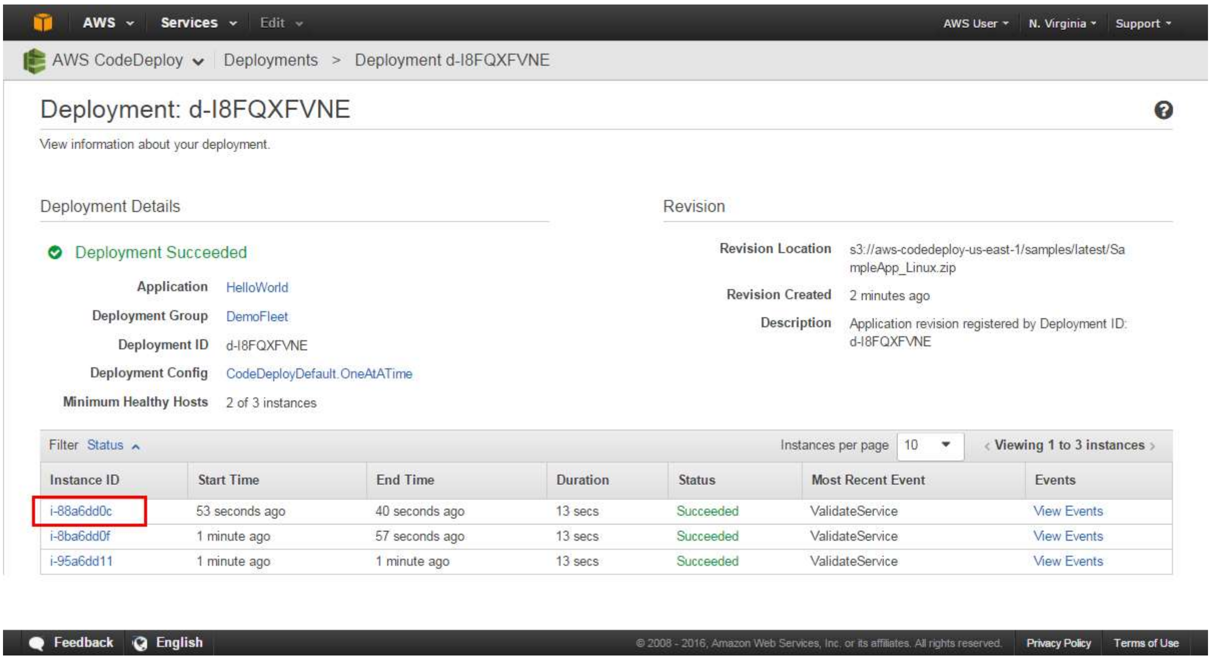
Instances

3 of 3 Instances Completed

3 Succeeded

View All Instances

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.


AWS
Services
Edit
AWS User
N. Virginia
Support
AWS CodeDeploy
Deployments > Deployment d-I8FQXFVNE
Feedback
English
© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Deployment: d-I8FQXFVNE

View information about your deployment.

Deployment Details

✓ Deployment Succeeded

Application: HelloWorld
Deployment Group: DemoFleet
Deployment ID: d-I8FQXFVNE
Deployment Config: CodeDeployDefault.OneAtATime
Minimum Healthy Hosts: 2 of 3 instances

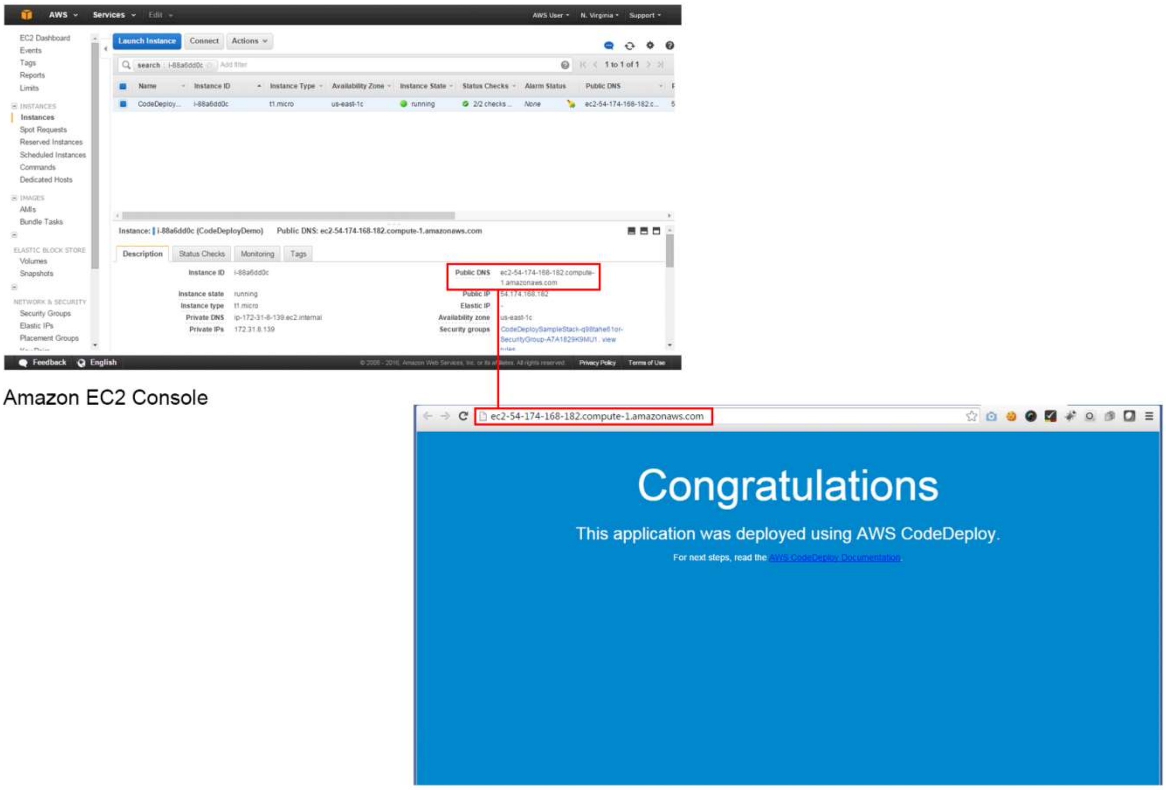
Revision

Revision Location: s3://aws-codedeploy-us-east-1/samples/latest/SampleApp_Linux.zip
Revision Created: 2 minutes ago
Description: Application revision registered by Deployment ID: d-I8FQXFVNE

Filter Status | Instances per page 10 | Viewing 1 to 3 instances

Instance ID	Start Time	End Time	Duration	Status	Most Recent Event	Events
i-88a6dd0c	53 seconds ago	40 seconds ago	13 secs	Succeeded	ValidateService	View Events
i-8ba6dd0f	1 minute ago	57 seconds ago	13 secs	Succeeded	ValidateService	View Events
i-95a6dd11	1 minute ago	1 minute ago	13 secs	Succeeded	ValidateService	View Events

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.



Instance: i-88a6dd0c (CodeDeployDemo) Public DNS: ec2-54-174-168-182.compute-1.amazonaws.com

Congratulations

This application was deployed using AWS CodeDeploy.

For next steps, read the AWS CodeDeploy Documentation

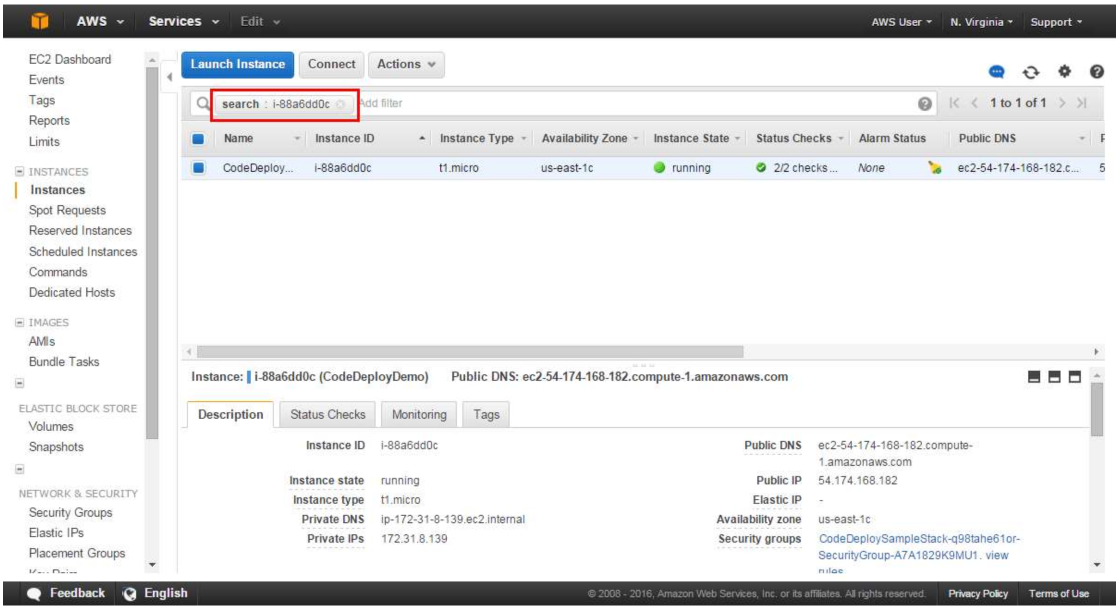
Amazon EC2 Console

Web Browser

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.



search : i-88a6dd0c

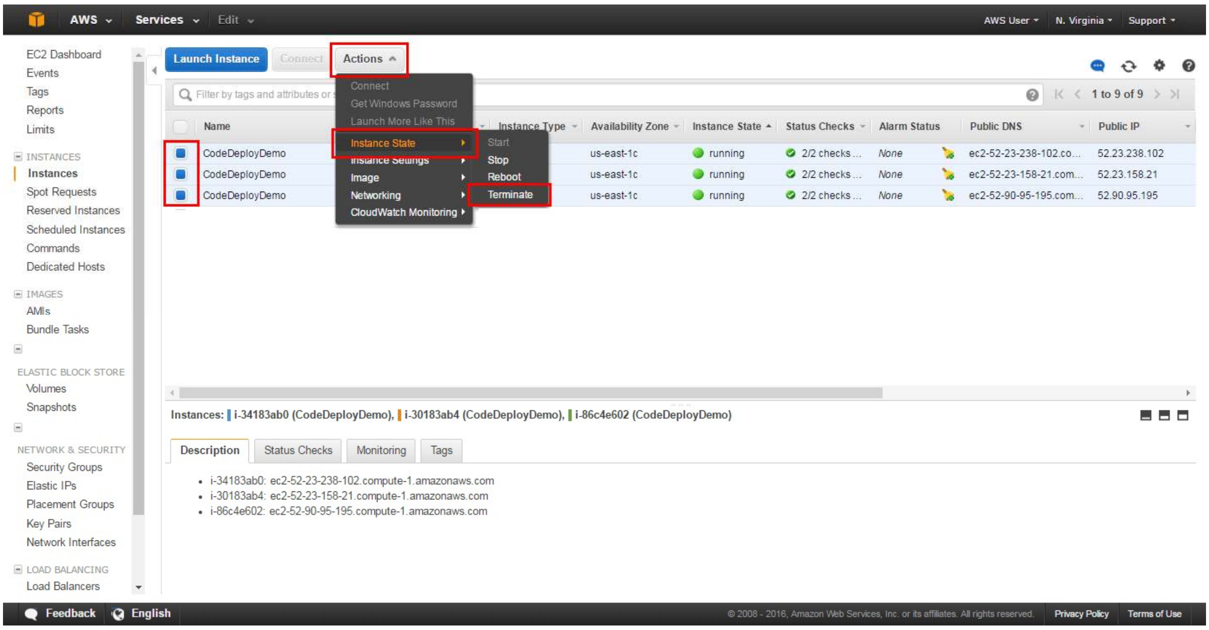
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
CodeDeploy...	i-88a6dd0c	t1.micro	us-east-1c	running	2/2 checks ...	None	ec2-54-174-168-182.c...

Instance: i-88a6dd0c (CodeDeployDemo) Public DNS: ec2-54-174-168-182.compute-1.amazonaws.com

Instance ID i-88a6dd0c
Instance state running
Instance type t1.micro
Private DNS ip-172-31-8-139.ec2.internal
Private IPs 172.31.8.139
Public DNS ec2-54-174-168-182.compute-1.amazonaws.com
Public IP 54.174.168.182
Elastic IP -
Availability zone us-east-1c
Security groups CodeDeploySampleStack-q98tahe61or-SecurityGroup-A7A1829K9MU1. view rules

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

When prompted, click Yes, Terminate.



	Name	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
■	CodeDeployDemo		us-east-1c	● running	✓ 2/2 checks ...	None	ec2-52-23-238-102.co...	52.23.238.102
■	CodeDeployDemo		us-east-1c	● running	✓ 2/2 checks ...	None	ec2-52-23-158-21.com...	52.23.158.21
■	CodeDeployDemo		us-east-1c	● running	✓ 2/2 checks ...	None	ec2-52-90-95-195.com...	52.90.95.195

Instances: i-34183ab0 (CodeDeployDemo), i-30183ab4 (CodeDeployDemo), i-86c4e602 (CodeDeployDemo)

Description | Status Checks | Monitoring | Tags

- i-34183ab0: ec2-52-23-238-102.compute-1.amazonaws.com
- i-30183ab4: ec2-52-23-158-21.compute-1.amazonaws.com
- i-86c4e602: ec2-52-90-95-195.compute-1.amazonaws.com