Linux Academy

Live! Lab

CodeDeploy Basics

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- Access to an AWS Console is provided on the Live! Lab page, along with your login credentials
- Ensure you are using the N. Virginia region
- Labs will automatically end once the alloted amount of time finishes

Related Courses

Manage & Deploy Code with AWS Developer Tools

Related Videos

Configure an EC2 Instance for Use with CodeDeploy

Create an Application, Deployment Group, and Deployment Configuration

Create and Configure the Appspec File

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Using files added to CodeCommit, we are going to create a CodeDeploy application. The *LiveLabRepo* from the previous lab has been created for you, and is available under the **CodeCommit Dashboard**. Similarly, under the **EC2 Dashboard**, the server, *LocalDevServer*, mimics the remote workstation we used in the prior lab.

Set Up and Configure an EC2 Instance

Before using CodeDeploy, we need to set up an EC2 instance to which we can deploy the code. Navigate to the **EC2 Dashboard**, under **Computer**. **Launch Instance**.

We want to use an *Amazon Linux AMI* with a type of *t2.micro*. On the **Configure Instance Details** page, set the **Auto-assign Public IP** option to *Enable*. For **IAM Role**, select the *codedeploy-basics* role that includes your username.

Finally, under **Advanced Details**, add the following Bash script:

```
#! /bin/bash
sudo yum update
sudo yum install ruby
sudo yum install wget
cd /home/ec2-user
wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto
```

This script installs general updates, Ruby, the AWS CLI and the CodeDeploy agent. When finished, click **Next: Add Storage**. Leave the default settings then press **Next: Tag Instance**.

We want to give the *Name* **Key** a **Value** of *LiveLabCodeDeployInstance*. **Next: Configure Security Group**.

Create a new security group. We want both *SSH* and *HTTP*. **Review and Launch**. Click **Launch** on the next screen.

A pop-up asks to select or create a key pair. Select *Create a new key pair* and give it a name. We used *livelabkeypair*. **Download Key Pair**. **Launch Instance**. Wait for the instance to finish provisioning, then use the login credentials (found under **Actions**, **Connect**) to SSH into your server using the appropriate key pair location:

```
    chmod 400 livelabkeypair.pem
    ssh -i “livelabkeypair.pem” ec2-user@ec2-54-161-176-217.compute-1.
amazonaws.com
```

Check if the CodeDeploy agent is running:

```
    [ec2-user@ip-10-0-0-13 ~]$ sudo service codedeploy-agent status
```

```
The AWS CodeDeploy agent is running as PID 2654
```

Our instance is now set up and correctly configured.

Create a CodeDeploy Application

With our instance launched and the CodeDeploy agent active, we can now create the CodeDeploy application, deployment group, and appspec file, and prepare the application for deployment.

Log out of your instance:

```
[ec2-user@ip-10-0-0-13 ~]$ logout
Connection to ec2-54-161-176-217.compute-1.amazonaws.com closed.
```

Return to the AWS Console, and navigate to the **CodeDeploy Dashboard**. Press **Get Started Now**. We want to work with a **Custom Deployment**. **Skip Walkthrough**.

Set the **Application Name** to *LiveLabApplication* and the **Deployment Group Name** to *LiveLabDeploymentGroup*.

Under **Add Instances**, we added the ***Amazon EC2*** instance with the **Key** *Name* of *LiveLabCodeDeployInstance*. Since we only have one instance, the **Deployment Configuration** can be set to anything; we left it at *CodeDeployDefault.OneAtATime*. Do not add **Triggers**.

Finally, for the **Service Role ARN**, select the pre-made *CodePipeline-Service* ARN. **Create Application**.

We now need to create the *appspec* file and package it with the files we want to deploy.

Configure the Appspec File

The files that we want to deploy are located on the development server that was already created at the start of the lab. Log into the *LocalDevServer* by retrieving the IP address from the EC2 Dashboard. SSH with the username *linuxacademy* and the password *123456*.

The directory we want to use is located in the */root/local-lab-repo* directory of the *root* user. Change to the root user and navigate to the directory:

```
[linuxacademy@ip-10-0-0-100 ~]$ su
[root@ip-10-0-0-100 linuxacademy]# cd /root/local-lab-repo
```

Two files, *newfile.txt* and *secondfile.txt* exist in this directory, much like in the previous lab. We now want to create the *appspec* file:

```
[root@ip-10-0-0-100 local-lab-repo]# touch appspec.yml
```

Open the file in your preferred text editor, and add the text as follows:

```
version 0.0
os: linux
files:
  - source: /
    destination: /home/linuxacademy/livelab
```

This file is extremely basic; in practice, your appspec file may have more information. Save and exit.

Upload Revision to S3 and Deploy

Return to the AWS Console, and navigate to the **S3 Dashboard**. **Create Bucket**. All buckets need unique names, so name yours as you see fit. We used *linuxacademy-livelab-333* and selected the **Region** as ***US Standard***. **Create**.

Return to the command line, and combine the files in the *local-lab-repo* into a zip file, upload it to the S3 bucket, and register it with CodeDeploy. This can be done in a single command:

```
    [root@ip-10-0-0-100 local-lab-repo]# aws deploy push --application-
name LiveLabApplication --description “Live lab deployment” --ignore-
hidden-files --s3-location s3://linuxacademy-livelab-333//deployme.zip
--source .
    To deploy with this revision, run:
    aws deploy create-deployment --application-name LiveLabApplication
--s3-location bucket=linuxacademy-livelab-333,key=/deployme.
zip,bundleType=zip,eTag=bec7b9c14a17236f4f71516dd4644c59 --deployment-
group-name <deployment-group-name> --deployment-config-name <deployment-
config-name> --description <description>
```

Insert the application name, description, and S3 location to match appropriate parameters.

Should you now refresh the S3 Dashboard, you can see the zip file just created.

Navigate back to the **CodeDeploy Dashboard** and select your application, then select the *LiveLabDeploymentGroup*. From here, choose **Deploy new revision** from under the **Actions** menu.

For the **Application** select *LiveLabApplication* and *LiveLabDeploymentGroup** for the **Deployment Group**. The **Revision Type** is *My application is stored in Amazon S3*. Choose the correct S3 file from the drop-down for the **Revision Location**. The rest can be left with default settings. **Deploy Now**.

Deployment may take a few minutes. When finished, log into the *LiveLabCodeDeploy* instance, and navigate to the */home/linuxacademy/livelab* directory. The files have now been deployed to the server!