[version_1.0]

Note

The exercises in this course will have an associated charge in your AWS account. In this exercise, you will create the following resources:

- Amazon Simple Storage Service (Amazon S3) bucket
- AWS CodeDeploy application and deployment group

The final exercise includes instructions to delete all the resources that you create in the exercises.

Familiarize yourself with **Amazon S3 pricing**, **AWS CodeDeploy**, and **AWS Free Tier**.

Exercise: Deploying AWS CodeDeploy Revisions

In this exercise, you will create an S3 bucket and upload the blog to that bucket. You will also set up the CodeDeploy group and make a revision that will then be deployed.

Task 1: Creating an S3 bucket and uploading the .zip file

In this task, you create an S3 bucket and upload the blog data to it.

1. Choose **Services**, then search for and open **EC2**.
2. In the navigation pane, choose **Instances**.
3. Confirm that the `TEST-environment` instance is running. If you need to restart the instance, choose **Instance state** and select **Start instance**.
4. Choose **Services**, then search for and open **S3**.
5. Choose **Create bucket**.
6. In **Bucket name**, enter a name. The bucket must have a unique name. As an example, you can use the following format: `devops-exercise2-<your_initials>-<random_number>` **Note:** If the bucket name you entered is already being used, try entering a different number.
7. Make sure that the **AWS Region** is **US West (Oregon) us-west-2**, which is the *same* Region where you will deploy your CodeDeploy group.
8. Choose **Create bucket**. The bucket you created should appear in the list of buckets.
9. Download the following file: Blog.zip

10. Back in the S3 console, choose the **Name** link for the bucket you created and then choose **Upload**.

11. Choose **Add files**, then browse to where you saved the `Blog.zip` file, open the file, and choose **Upload**.

12. Choose **Close**.

Task 2: Creating and configuring the CodeDeploy application

In this task, you create and configure the CodeDeploy application.

1. Choose **Services**, and search for and open **CodeDeploy**.

2. In the navigation pane under **Deploy**, choose **Applications**.

3. Choose **Create application** and configure the following settings.
    - **Application name**: `TestApplication`
    - **Compute platform**: *EC2/On-premises*

4. Choose **Create application**.

5. Choose **Create deployment group** and configure the following settings.
    - **Deployment group name**: `TestDeploymentGroup`
    - **Service role**: `CodeDeployServiceRole`
    - **Deployment type**: Keep *In-place* selected
    - **Environment configuration**: *Amazon EC2 instances*
        - **Key**: `Name`
        - **Value**: `TEST-environment`
    - **Agent configuration with AWS Systems Manager > Install AWS CodeDeploy Agent**: *Never*
    - **Deployment settings**: Keep *CodeDeployDefault.AllAtOnce* selected
    - **Load balancer**: Clear *Enable load balancing*

6. Choose **Create deployment group**.

7. Choose **Create deployment** and configure the following settings.
    - **Deployment group**: Keep *TestDeploymentGroup* selected
    - **Revision location**: Use the S3 bucket and Blog.zip file that you created previously (example URL format: `s3://devops-exercise2-<your_initials>-<random_number>/Blog.zip`)
    - **Revision file type**: *.zip*
    - **Deployment description**: Feel free to add a description

8. Choose **Create deployment**.

9. In the **Deployment lifecycle events** section, choose the **Instance ID** link.

10. Select the `TEST-environment` instance.

11. Copy the **Public IPv4 address** or **Public IPv4 DNS** address and paste it in a new browser tab.

You should now see the DevOps blog.

Task 3: Uploading the new revision

In this task, you edit some of the code for the blog website. You then deploy the new changes to your bucket and the CodeDeploy group.

1. On your local computer, extract the `Blog.zip` file.
2. In a text editor, open the `index.html` file.
3. Make two changes to the `index.html` file by changing the values for ***background-color*** and ***Version***.
   - Locate `background-color: #0000FF;` and change it to `background-color: #FFC0CB;`
   - Locate `Version 1` and change to `Version 2`
4. Save the file and re-compress the files into a `Blog.zip` file.
5. Back in the AWS Management Console, search for and open **S3**.
6. Under **Buckets**, choose the link for the bucket you created in Task 1.
7. Select `Blog.zip` and choose **Delete**.
8. In the **Permanently delete objects?** section, enter `permanently delete` and then choose **Delete objects**.
9. Choose **Close**.
10. Choose **Upload**, and then choose **Add files**.
11. Browse to the edited `Blog.zip` file that's on your local computer, and open it.
12. Choose **Upload**.
13. Choose **Services**, and search for and open **CodeDeploy**.
14. In the navigation pane, choose **Deploy > Application** and choose the **TestApplication** link.
15. Choose the **TestDeploymentGroup** link, choose **Create deployment**, and configure the following settings:
    - **Revision location**: Again, use the S3 location that you used for the previous deployment (example format: `s3://devops-exercise2-<your_initials>-<random_number>/Blog.zip`)
    - **Revision file type**: *.zip*
    - **Deployment description**: `Deploys version 2 of the application to the TEST environment`
16. Choose **Create deployment**.
17. Under **Deployment lifecycle events**, choose the **Instance ID**.
18. Select the `TEST-environment` instance.

19. Copy the **Public IPv4 address** or **Public IPv4 DNS** address and paste it into a new browser tab.

You should now see the DevOps blog with the new revisions.