[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
- AWS CodeCommit repository
- AWS Identity and Access Management (IAM) roles
- Amazon Elastic Compute Cloud (Amazon EC2) instance
- AWS CodePipeline pipeline

This final exercise includes instructions to delete all the resources that you created in the course exercises.

Familiarize yourself with **Amazon S3 pricing**, **AWS CodeDeploy pricing**, **AWS CodeCommit pricing**, **AWS CodePipeline pricing**, **Amazon EC2 pricing**, and the **AWS Free Tier**.

Exercise: Working with AWS CloudFormation and AWS CodePipeline

In this exercise, you will use a provided CloudFormation template to set up the backend services such as CodePipeline, CodeCommit, CodeDeploy, and CodeBuild. You will then upload new revisions to the pipeline.

Task 1: Setting up the backend with CloudFormation

In this task, you deploy a CloudFormation stack that will set up the backend architecture for this exercise.

1. Download the following CloudFormation template: final_pipeline.yaml
2. In the AWS Management Console, choose **Services**, and search for and choose **CloudFormation**.
3. In the CloudFormation console, choose **Create stack** and choose **With new resources (standard)**.
4. In the **Specify template** page, select **Upload a template file** and select **Choose file**.
5. Upload the `final_pipeline.yaml` file by opening the directory where you downloaded it, choosing the `final_pipeline.yaml` file, and choosing **Open**.
6. Choose **Next**.
7. In the **Specify stack details** page, enter the following values.

- **Stack name**: `final-pipeline`
- **Parameters > CodePipelineName**: `final-pipeline`

1. Choose **Next**.

2. Skip **Configure stack options** by choosing **Next**.

3. At the bottom of the **Review final-pipeline** page, select **I acknowledge that AWS CloudFormation might create IAM resources with custom names** and choose **Create stack**.

Creating the stack can take some time to complete. Allow the stack to finish before you continue to the next task.

The template will create a CodeDeploy application (`CodePipeBlogSampleApplication`), and a deployment group (`MyDemoDeploymentGroup`). The stack also contains the following resources: * CodeCommit repository (`NewsletterRepo`) * CodePipeline pipeline (`final-pipeline`) * S3 bucket for the pipeline artifacts * EC2 instance (`CodePipelineBlog`) * CodeBuild project (`NewsletterBuild`)

Task 2: Viewing deployment events in CodePipeline

In this task, you view the deployment events in the pipeline.

1. In the console, use the **Services** search box to find and open **AWS CodePipeline**.

2. Under **Pipelines**, choose `final-pipeline` .

3. Under **Deploy**, choose **Details**.

4. In the details page, scroll to the bottom, and under **Deployment lifecycle events**, choose the **Instance ID**.

5. Open the instance details pane by selecting the check box for the `CodePipelineBlog` instance.

6. In the **Instance summary** page, copy the **Public IPv4 DNS** address and paste it in a browser.

You should see the blog, which was deployed automatically using the CloudFormation template.

1. Switch back to the **CodeDeploy** tab.

2. In the navigation pane, expand **Pipeline** and choose **Pipelines**.

3. Choose the `final-pipeline` link.

4. Under **AcceptanceTest**, choose **Details**.

Review the details and note all the acceptance tests (which are in blue). These tests were used to verify that the application was deployed and tested successfully.

Task 3: Committing a new revision

In this task, you make a new revision to the blog website and commit those changes.

1. Use the **Services** search box to find and open **CloudShell**. Wait for the shell to initialize.
2. Inside the shell, run the following commands:

   Configure Git and clone the repository:

   ```
   git config --global user.email "YOUR_EMAIL"
   git config --global user.name "YOUR_NAME"
   git clone codecommit://NewsletterRepo
   ```

   Change the color of the horizontal rule (`<hr>` HTML tag) and the blog version number:

   ```
   cd NewsletterRepo/
   sed -i 's/#0000FF/#800080/g' index.html
   sed -i 's/Version 1/Version 2/g' index.html
   ```

   View the changes:

   ```
   git diff
   ```

   Create a commit and push the changes:

   ```
   git add *
   git commit -m "version 2 new hr color"
   git push
   ```

3. In the console, search for and open **CodePipeline**.

You should see a pipeline execution that was started by the new commit.

1. View the details by choosing the link for the most recent pipeline execution.

The pipeline execution might take a few minutes to finish.

1. After `AcceptanceTest` is complete, search for and open **EC2**.
2. If you are not in the **Instances** page, from the navigation pane, expand **Instances** and then choose **Instances**.
3. Open the instance details section by selecting the check box for the `CodePipelineBlog` instance.
4. Copy the **Public IPv4 address** and paste it in a new browser tab.

You should see the new changes that you made reflected in the blog.

Deleting all exercise resources

Congratulations! You have successfully completed the course project. In this task, you delete the AWS resources that you created for this project.

1. Open the **Amazon S3** console.

- Empty and delete the bucket that you created for the previous exercise. It might have a name similar to: **devops-exercise2-<your_initials>-<random_number>**
- Empty and delete the bucket with the name: **final-pipeline-codepipelineartifactstores3bucket-<random_string>**.

1. Open the **AWS CloudFormation** console.

- Delete the **final-pipeline** stack.

1. Open the **Amazon EC2** console.
    - Terminate the **TEST-environment** instance that you created in the previous exercise.
2. Open the **CodeDeploy** console, and from the navigation pane, choose **Applications**.

- Delete **TestApplication** that you created in the previous exercise.