[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 exercise includes instructions to delete all the resources that you created for this exercise.

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

Exercise: Logging

In this exercise, you will use AWS Systems Manager to run commands. These commands will use the grep utility to search Apache logs on the underlying instance. You will then output these logs to Amazon CloudWatch.

Task 1: Setting up with AWS CloudFormation

In this task, you use the same CloudFormation stack that you used in the previous 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 navigation pane, choose **Stacks**.
4. Choose **Create stack** and choose **With new resources (standard)**.
5. In the **Specify template** page, select **Upload a template file** and select **Choose file**.
6. Upload the `final_pipeline.yaml` file by opening the directory where you downloaded it, choosing the `final_pipeline.yaml` file, and choosing **Open**.
7. Choose **Next**.

8. In the **Specify stack details** page, enter the following values.
    - **Stack name**: `final-pipeline`
    - **CodePipelineName**: `final-pipeline`
9. Choose **Next**.
10. Skip **Configure stack options** by choosing **Next**.
11. 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.

Task 2: Searching for information in log files

In this task, you use grep to search the Apache logs on the underlying EC2 instance. To do so, you use AWS Systems Manager to run some shell commands.

1. Choose **Services**, and search for and open **SSM**.
2. In the navigation pane, under **Node Management**, choose **Run Command**.
3. Choose **Run a Command**.
4. In the **Command document** search box, enter `script` and press Enter. *Wait for the command document options to load*.
5. Select **AWS-RunShellScript**.

   You are looking for any 404 errors that occurred and are logged in the Apache log file.
6. In the **Command parameters** box, paste the following command:

   ```
   cat /etc/httpd/logs/access_log | grep 404
   ```
7. In the **Targets** section, add a tag by keeping **Specify instance tags** selected, entering the following values, and then choosing **Add**:
    - **Tag key**: `Name`
    - **Tag value**: `CodePipelineBlog`
8. Under **Output options**, configure the following settings.
    - **Enable an S3 bucket**: Clear this check box
    - **Enable CloudWatch logs**: Select this check box
    - **Log group name**: `/runcommand/404-check`
9. Choose **Run**.

In the **Targets and outputs** section, the status should eventually show as *Success*.

1. To see the log, you can drill down by choosing the instance under **Instance ID** and then expanding **Output**.

Task 3: Viewing the Apache logs with Amazon CloudWatch Logs

In this task, you also view the Apache logs by using Amazon CloudWatch Logs.

1. Open the **CloudWatch** console.
2. In the navigation pane, under **Logs**, choose **Log groups**.

   In the **Log group** list, you should see */runcommand/404-check*.
3. Choose the **/runcommand/404-check** link.
4. Choose the **Log stream**.

   You should see the 404 error log that was also in the output from the previous task.

Cleaning up

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 with the name: **final-pipeline-codepipelineartifactstores3bucket-<random_string>**.
2. Open the **AWS CloudFormation console**.
   - Delete the **final-pipeline** stack.
3. Open the **CloudWatch** console and in the navigation pane under **Logs**, choose **Log groups**.
   - Delete the **/runcommand/404-check** group.