IAM Users, Roles & Policies

Add IAM roles & policies needed to meet the assignment objectives to the CloudFormation Template.

Application Logging & Metrics

AMI Updates

1. Update your packer template to [install](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-on-first-instance.html#download-CloudWatch-Agent-on-EC2-Instance-first) the [Unified CloudWatch Agent](https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/UseCloudWatchUnifiedAgent.html) in your AMIs.
2. Your CloudWatch agent must be set up to start automatically when an EC2 instance is launched using your AMI.
3. Note that until CodeDeploy deploys the application on the EC2 instance, the CloudWatch agent may not work.
   1. You should restart the CloudWatch agent in your after install CodeDeploy lifecycle hook.

IAM Updates

* Update the CloudFormation template to [Update the IAM role attached to the EC2 instance for use with CloudWatch Agent](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create-iam-roles-for-cloudwatch-agent.html).

WebApp User Stories

1. As a user, I want all application log data to be available in CloudWatch.
2. As a user, I want metrics on API usage available in CloudWatch.
3. Create the following custom metrics for every API we have implemented in the web application. The metrics data should be collected in CloudWatch.
   1. Count the number of times each API is called.
4. You can retrieve custom metrics using either [StatsD](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Agent-custom-metrics-statsd.html) or [collectd](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Agent-custom-metrics-collectd.html).
5. CloudWatch agent configuration file must be copied over to the EC2 server when the application is being deployed by CodeDeploy.
6. (RE) Configure the CloudWatch agent before starting your service in the *afterInstall* hook of the CodeDeploy lifecycle.

Sample CloudWatch Agent Config

{  
  "agent": {  
      "metrics\_collection\_interval": 10,  
      "logfile": "/var/logs/amazon-cloudwatch-agent.log"  
  },  
  "logs": {  
      "logs\_collected": {  
          "files": {  
              "collect\_list": [  
                  {  
                      "file\_path": "/var/log/tomcat9/csye6225.log",  
                      "log\_group\_name": "csye6225-fall2020",  
                      "log\_stream\_name": "webapp"  
                  }  
              ]  
          }  
      },  
      "log\_stream\_name": "cloudwatch\_log\_stream"  
  },  
  "metrics":{  
    "metrics\_collected":{  
       "statsd":{  
          "service\_address":":8125",  
          "metrics\_collection\_interval":15,  
          "metrics\_aggregation\_interval":300  
       }  
    }  
 }  
}

Command to Configure CloudWatch Agent

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \

-a fetch-config \

-m ec2 \

-c file:/opt/cloudwatch-config.json \

-s

Documentation

CloudWatch Logs

* [Create IAM Roles and Users for Use With CloudWatch Agent](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create-iam-roles-for-cloudwatch-agent.html)
* [Install CloudWatch Agent Package on an Amazon EC2 Instance](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/install-CloudWatch-Agent-on-first-instance.html#download-CloudWatch-Agent-on-EC2-Instance-first)

CloudWatch Metrics

* [Retrieve Custom Metrics with StatsD](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Agent-custom-metrics-statsd.html)
* [Retrieve Custom Metrics with collected](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Agent-custom-metrics-collectd.html)

Submission

The assignment will be considered late if commits are made to the **main** branch after the due date.

1. All work for this assignment must be done on the feature branch in your fork and merged to main when you are dev complete.
2. The feature and main branches must be in sync.
3. Submit your code from **all** repositories in this assignment. **Read the instructions carefully to create your zip file.**
   1. Create a folder with the naming convention **firstname\_lastname\_neuid**
   2. In the **firstname\_lastname\_neuid**clone all of your GitHub (organization) repositories with the **git clone**command. It is important that you clone the repos so that your commit history and branches are preserved.
   3. Once you have cloned all of your repositories, you will create a zip of the **firstname\_lastname\_neuid\_a\_##** directory. The zip file should be **firstname\_lastname\_neuid\_a\_##.zip** where **##** is the assignment number.
   4. Now unzip the zip file in some other directory and confirm the content of the zip files.
   5. Upload the Zip to this assignment.
4. You are allowed to resubmit. If you think there may be an issue with the ZIP file, feel free to submit it again. Only the latest will be used grading.

Grading Guidelines

The following guidelines are for information only. They are subject to change at the discretion of the instructor and TA. 10% penalty will be imposed if the application crashes due to unhandled exceptions/errors.

Previous Assignment Objectives

* TAs must verify that students have resolved all open issues from the previous assignment(s).

Git Repository Content Check (10% Penalty)

* Check the repository for any IDE-specific files. IDE configuration files must not be in the repository.
  + Verify their .gitignore configuration.
* Check the repository for build artifacts such as .class, .jar, .war files and build, node\_modules directory. None of these should be checked into the repository.
* Check for dependencies. Dependencies from the Maven repository or npm should not be committed to the git repository.

Pre-requisite for Demo (5% Penalty)

* Delete all VPCs you may have created in your demo AWS account.

Application Crash (10% Penalty)

* The application should not crash or return *500 Internal Server Errors* during the demo.

Security Check (15% Penalty)

* Check for hard-coded AWS credentials.
* Check IAM policies that grant access to all (\*) resources.
* Check for IAM roles and policies that have AdministratorAccess attached to them.
* Image deletion must be hard deleted in the S3 bucket.

CloudWatch Logs & Metrics (100%)

1. Students to demo application logs in CloudWatch.
   1. Monitor the logs as a new version of the application is deployed. Look for logs generated in the last 5 minutes.
   2. Look for log messages generated by the application from API calls made for the demo.
   3. As there may be some lag before logs are available in CloudWatch, wait for about 2-5 minutes for data to be available.
2. Students must demo application metrics that are being pushed to CloudWatch.
   1. Verify metrics for counters and timers as you make API calls for the demo.
   2. Note the value of these metrics before and after the demo.
   3. Count the number of API calls being made and see if the count displayed in the dashboard adds up.
   4. When looking at metrics in the dashboard, look for data from the last 5 or 15 minutes.
   5. As there may be some lag before metrics are available in CloudWatch, wait for about 1-2 minutes for data to be available.