Individual Project: AWS Basics

Due Jun 8 by 11:59pm **Points** 100 **Submitting** a file upload **Available** until Jun 10 at 11:59pm

This assignment was locked Jun 10 at 11:59pm.

Individual Project 1: AWS Basics

Due by Jun 8th 11:59 pm AZ time

Summary

In the first project, we will build an application that can communicate with IaaS service online and request certain resources from the IaaS provider. Specifically, we will build this application using the Amazon Web Services (AWS) SDK and API. AWS is the most widely used IaaS provider and offers a variety of computing, storage, and messaging services. The technologies and techniques that we learn during this project will be useful for us to build others in the future.

Description

This is an individual project.

You need to program an application that requests a list of required resources from AWS, check and list all the resources assigned to you, then destroy all the resources.

The required resources are:

1. An EC2 instance(VM):
   a. Use free Ubuntu Image
   b. T2.micro tier
   c. Using your own key pair with your name as the key pair name.
2. An S3 bucket.
3. An SQS queue with the FIFO type.

Your application should:

1. Load the AWS SDK(you may use whatever language you prefer, as long as there is an AWS SDK available)
2. Read your access information, to be able to access AWS service (Guidance of how to get access information like access key ID, token and session key are included in the AWS Introduction ppt that was shared with you earlier.)
3. Send resource request API call to AWS to create the EC2 instance, S3 bucket, and SQS queue.
4. Wait for 1 min.
5. List all EC2 instances, S3 buckets, and SQS queues in your accounts in the current region again, print them out.
6. Upload an empty text file with the name “CSE546test.txt” into the S3 bucket that you just created.
7. Send a message with the message name “test message” and message body “This is a test message” into the SQS queue.
8. Check how many messages are there in your SQS queue, print it out in a new line.
9. Pull the message you just sent from the SQS queue, print out the message title and body in two lines.
10. Check how many messages are there in your SQS queue again.
11. Delete all the resources.
12. List all EC2 instances, S3 buckets, and SQS queues in your accounts in the current region again.
13. Also, print out a message each time an action is done. For example, when you sent all the resource requests and start to wait for 1 minute, you should print out “Request sent, wait for 1 min”. Another example, after you sent the message to SQS queue, you should print out a new line: “Message sent”. And so on.

Test your code thoroughly. Check the following:

All the resources are correctly created, 1 minute should be more than enough for AWS to create them, you may increase the waiting time during your test;

The file is uploaded to S3 bucket;

SQS queue received your message, stored and sent it.

Your project will be graded according to the above criteria.

Submission

You need to submit your implementation on Canvas by June 8th 11:59 pm AZ time. Your submission should be a single zip file that is named “CSE546_” plus your full names. The zip file should contain all your source code (For languages like Java or C++, you should also provide extra complied/binary files, like .jar file). It should also contain a simple README file of how to run your application. **Please includes screenshot/s of the output of your program and attach them into the README file.**

Failure to follow the above submission instructions will cause a penalty to your grade. The submission link will be closed immediately after the deadline.

Policies

Late submissions will absolutely not be graded (unless you have verifiable proof of emergency). It is much better to submit partial work on time and get partial credit for your work than to submit late for no credit.

We encourage high-level discussions among students to help each other understand the techniques. However, code-level discussion is prohibited and plagiarism will directly lead to the failure of this course. We will use anti-plagiarism tools to detect violations of this policy.

◂ Previous Next ▸

Submission

Submitted!
Jun 6 at 2:50pm
Submission Details
Download CSE546_Debesh Mishra.zip

Grade: 100 (100 pts possible)
Graded Anonymously: no

Comments:
No Comments