Project 0: Onboarding

Due by **02/02/2025 11:59PM**

Summary

This warmup project will help us familiarize ourselves with AWS development and our project submission and grading process. Specifically, the project will 1) provide you the development environment that you will use for all the following projects; and 2) help you understand how to test and submit your project properly.

We recommend you to follow the steps below to complete the project. But these are not exact step-by-step instructions. Check the relevant documentation and ask on Ed Discussion if you have questions.

AWS Development Setup

1. **Setting up the AWS Account**
    a. Go to this link to create an AWS account.
    b. Choose Account Type as "Personal" and fill in your Contact and Payment details
    c. In the Identity Verification step, confirm your identity.
    d. Select the "Basic Plan"

    Important: Make sure to monitor your AWS usage and billing so you do not get charged. Use only the resources from the AWS Free Tier. Follow this tutorial to use the CloudWatch to automatically generate billing alerts to you.

2. **How to get AWS Access Key ID**
    a. Go to the Amazon Web Services console and click on the name of your account (it is located in the top right corner of the console).
    b. In the expanded drop-down list, select *Security Credentials*.
    c. Click the "*Generate new key*" in "*Access keys* (access key ID and secret access key)".
    d. Click "*Show key*", and you will see the Access Key ID and the Secret Access Key.
    e. Copy your Access Key ID and the Secret Access Key to Eclipse for configuring the AWS account on the editor.

3. **Creating IAM users for development and grading**
    An AWS Identity and Access Management (IAM) user is an entity you create in AWS. The

IAM user represents the human user or workload who uses the IAM user to interact with AWS. A user in AWS consists of a name and credentials. An IAM user with administrator permissions differ from an AWS account root user.

We ask you to create two IAM users, one used by yourself for the development of your cloud app; and the other for the TA to use to check and grade your app. Follow this guide to set up IAM users. Follow the least privilege principle to give each IAM user only the necessary permissions.

- **Development IAM**
  - Use this for all your development tasks.
  - Assign full permissions for any AWS resource you use for this project.
- **Grading IAM**
  - The TA will use this only for grading.
  - You **MUST** name the grading IAM user as "**cse546-AutoGrader**"
  - For **Project-0,** the Grading IAM requires only these permissions
    - AmazonEC2ReadOnlyAccess
    - AmazonS3ReadOnlyAccess
    - AmazonSQSReadOnlyAccess
    - IAMReadOnlyAccess

Submission Requirements & Guidelines

1. Submit the following in a zip file to Canvas:
   a. Credentials: Make a directory named '**credentials'**, and include your txt file named '**credentials.txt**'
   b. In the '**credentials'** folder, create a txt file "**credentials.txt**" with following parameter values separated by commas in the order as mentioned below
      i. ACCESS KEY ID of the grading IAM user
      ii. SECRET ACCESS KEY of the grading IAM user

```
kjha9@en4113732l:~/git/GTA-CSE546-SPRING-2025/Project-0/grader/credentials$ cat credentials.txt
AKIA5GLEVSDVHWPNQX5H,dygP3gamUan3YAYHQj2RO52zM68OMn/HIV1XFRGl
kjha9@en4113732l:~/git/GTA-CSE546-SPRING-2025/Project-0/grader/credentials$
```

2. You **MUST** name the zip file following the below naming convention: "Project0-<Your ASU ID>.zip", e.g., Project0-1225754101.zip

   You can use the following command to create your zip file:

```
zip -r Project0-1225754101.zip credentials/
```

Note: Extensions automatically generated by Canvas due to multiple submission attempts will be managed by the grading script.

3. Do not submit any other files

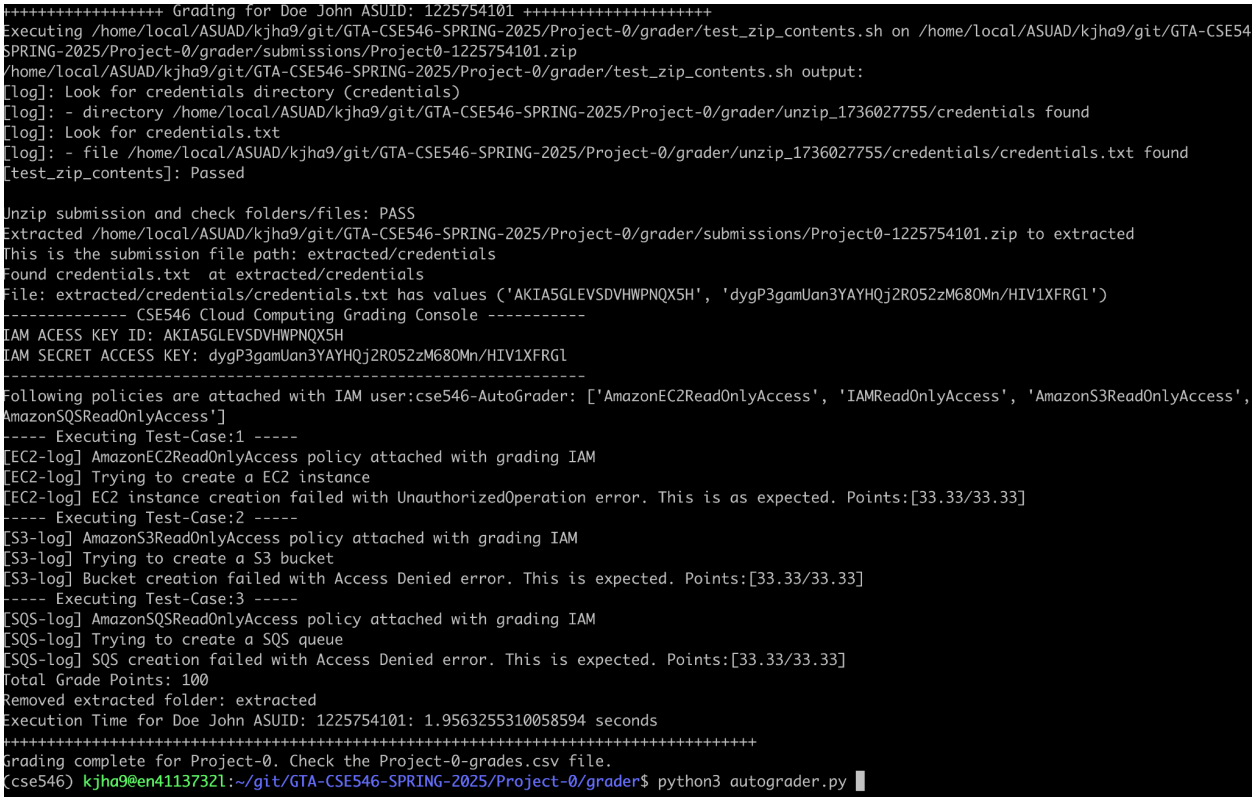
Testing

Make sure that you use the provided autograder and follow the instructions below to test your project submission. Failure to do so may cause you to lose all the project points and there will be absolutely no second chance.

1. Download the zip file that you submitted from Canvas.
2. Download the autograder from GitHub:
   https://github.com/CSE546-Cloud-Computing/CSE546-SPRING-2025.git
   In order to clone the GitHub repository follow the below steps:
   a. git clone https://github.com/CSE546-Cloud-Computing/CSE546-SPRING-2025.git
   b. cd CSE546-SPRING-2025/
   c. git checkout project-0
   d. Create a directory "**submissions**" in the CSE546-SPRING-2025 directory and move your zip file to the submissions directory.
2. Prepare to run the autograder
   a. Install Python: `sudo apt install python3`
   b. Populate the class_roster.csv
      i. If you are a student; replace the given template only with your details.
      ii. If you are a grader; use the class roster for the entire class
3. Run the autograder
   - Run the autograder: `python3 autograder.py`
   - The autograder will look for submissions for each entry present in the **class_roster.csv**
   - For each submission the autograder will
     - Validate if the zip file adheres to the submission guidelines as mentioned in the project document.
       - If Yes; proceed to next step
       - If No; allocate 0 grade points and proceed to the next submission
     - The autograder extracts the credentials.txt from the submission and parses the entries.
     - Use the Grader IAM credentials to test the project as per the grading rubrics and allocate grade points.
   - The autograder will dump stdout and stderr in a log file named **autograder.log**

Note: The autograder is case sensitive and space sensitive when looking for the zip file and the files and directories. If it cannot find your zip file or the required files or directories, you will lose a lot of points per the grading rubrics.
Note: Use a Linux-based system to run the autograder. We recommend you to install a Linux VM if you are using Windows/Mac.

4. Sample Output



Grading Rubrics

	Test Case	Test Criteria	Points
0	Unzip submission and check folders/files	1. Submission can be unzipped 2. Required folders/files can be found 3. No extra files	1. Deduct all points if unzip fails 2. Deduct all points if credentials.txt can't be found 3. Deduct 10 if there are extra files and another 10 if extra binary files

1	Validate IAMReadOnly access policy is attached to IAM user	IAMReadOnlyAccess must be attached to the grading IAM user	Deduct all points if IAMReadOnlyAccess is not attached to the grading IAM user
2	Validate the permissions to access EC2	1. AmazonEC2ReadOnlyAccess policy is attached to the grading IAM user and 2. EC2 instance creation fails with 'UnauthorizedOperation' exception	33.3
3	Validate the permissions to access S3	1. AmazonS3ReadOnlyAccess policy is attached to the grading IAM user and 2. S3 bucket creation fails with 'AccessDenied' exception.	33.3
4	Validate the permissions to access SQS	1. AmazonSQSReadOnlyAccess policy is attached to the grading IAM user and 2. SQS queue creation fails with 'AccessDenied' exception.	33.3

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.
- Every student needs to **work independently** on this exercise. We encourage high-level discussions among students to help each other understand the concepts and principles. However, a code-level discussion is prohibited and plagiarism will directly lead to failure of this course. We will use anti-plagiarism tools to detect violations of this policy.
- **The use of generative AI tools is not allowed** to complete any portion of the assignment.