Michael Soltys
michael.soltys@csuci.edu

Software Project
"Submitter"

COMP 350 - Spring 2021
Last Updated: Jan 13, 2021

In the project you will design and implement an application which receives Python code, runs it against pre-assigned test cases, and grades it. The application will be written using the AWS cloud infrastructure, and it will be written in the style of de-coupled microservices. The use cases of this application will be programming competitions, course assignment submissions and other coding events.

There are going to be two user types of this application: *Student* and *Administrator*.

The Admin will start by either registering a new event (say a course or competition) or modifying an existing event. In order to register a new event, the following info should be included:

1. Event name, e.g., 'COMP350' or 'ACM 2021 regional prep'
2. Dates (start and end)
3. Max and min members per team, e.g., '1-1' if assignments to be done individually, or '2-3' if in teams of at least 2 but no more than 3, or 'all' for one team consisting of all participants
4. Assignment, e.g., 'assignment 1'; note that a particular event may have several assignments associated with it, this is natural both for a course that has more than one assignment, or a competition where there is usually more than one problem
5. Assignment due date/time
6. Test cases and Grade cases
7. List of student emails

Once this is submitted, the app will generate a token for each student. A token will be a unique identifier / password (in the sense that it will *authenticate* a given student, and *authorize* the student to submit a file for a given (event, assignment) pair). For example, the token could be computed from the concatenation of: Admin password + Event Name + Assignment Name + Student email. A token should be generated with MD5, e.g.,

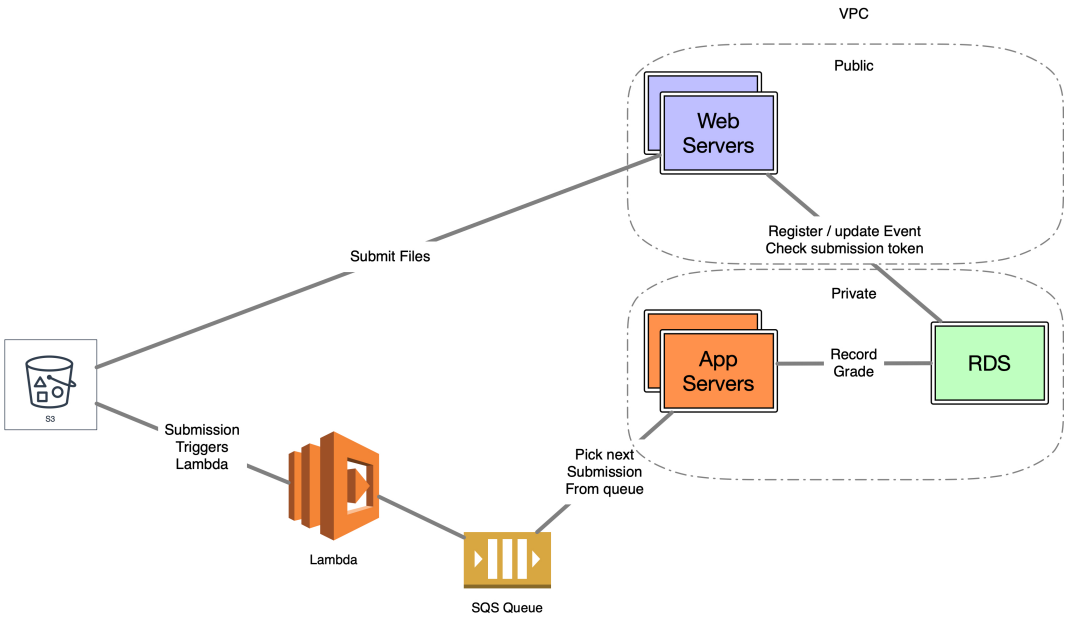
```
6708d147bc516f040ad97fcb73e28964
```

The system should then email the tokens to the students (using AWS SNS).

If the Admin wants to modify an existing event, what should be done if submissions have already been made?

There should be a very simple "Submissions Web Page" with fields for tokens. A student, or group of students, will make a submission by entering their tokens (if the students are working in a team, several tokens will be entered), and that will open the page for submission. This means that the student(s) can now upload their Python code file, and two things will happen: (i) there will be an output of running the program on the test cases, and (ii) the assignment will be run on grade cases without giving output and the grade recorded. Students may choose to re-submit based on the test cases output; re-submissions should be allowed until the due date/time.

Technical Requirements: The app should work in a dedicated VPC. The VPC should have two subnets, one Public and one Private. The Public subnet will house EC2 instances responsible for running the web portal for submission. All submissions should be saved in an S3 bucket. When a submission file is saved in an S3 bucket, a λ function should be triggered which will place a message in an SQS queue. The message will be consumed by an EC2 instance in the Private subnet responsible for running the file on test and grade cases. The results should be recorded in an RDS in the Private subnet. The S3 bucket should be connected to the VPC via an endpoint.



Notes: keep all old versions of submissions, save on storage by having a lifecycle rule that moves them to Glacier storage. Use credential management, i.e., do not "hardwire" credentials in the code. Use GitHub to manage the project (instructor will provide private repository). Build a VPC. Use a budget; the project is budgeted $1,500, and you should stay within this budget. The execution of arbitrary code (when executing the submission) is dangerous; how to "sandbox it"? The grades should be exportable in CSV.