

write-up

Periodically checks the completion times of jobs belonging to Free users and moves their results files from the gas-results bucket to the gas-archive bucket.

I use the long polling periodically listens to the archive and restore message from the message glacier SQS. The polling approaches are simpler but less scalable, so I send the glacier message data with the attribute 'type' to check whether we should 'archive' or 'restore' the job to make it scalable.

When annotating the job, in **run.py**, not only update the ['job\_status'] from 'RUNNING' to 'COMPLETED' to the result topic, it also publishes a glacier topic with type 'archive'(['type'] = 'archive'), and send the message to glacier message queue.

If the glacier message is 'archive', in **glacier\_updown.py**: if the user is premium user, then delete this message, since there is no need for premium user to archive the file. If the users are free users and have completed their annotation job more than 2 hours, then we will archive the job and change the job\_status in 'ARCHIVE'(different in homework 6, I add one more job\_status type). If the job\_status is "COMPLETED", then the free user can download the file, if the job\_status is "ARCHIVE", the free user can not download the file. Until they change their plan.

And in the **mpcs\_app.py**, when the free\_user is subscribe to a premium user, then public a 'restore' topic to the glacier message queue, in the 'subscribe'. And we we update the user's role, change all the job files with this username from "ARCHIVE " to "COMPLETED"(['job\_status'] = "COMPLETED") and public 'restore' type queue to glacier sns.

If the glacier message is 'restore', **glacier\_updown.py**: if the job file is not Glacier or the object does not have a completed or ongoing restoration request, when we restore it.

Problem I met, some are dump questions:

1. when the job is running, my code will occur complete\_time error, since the job is not completed, so no attribute for complete\_time how to show different status when the job is running or completed ! I need to handle the message
2. # S3 Object  
obj = s3.Object(bucket\_name='boto3', key='test.py')  
response = obj.get()  
data = response['Body'].read()
3. how to deal with the buffer message! in SQS console, delete !
4. auto scaling configuration !