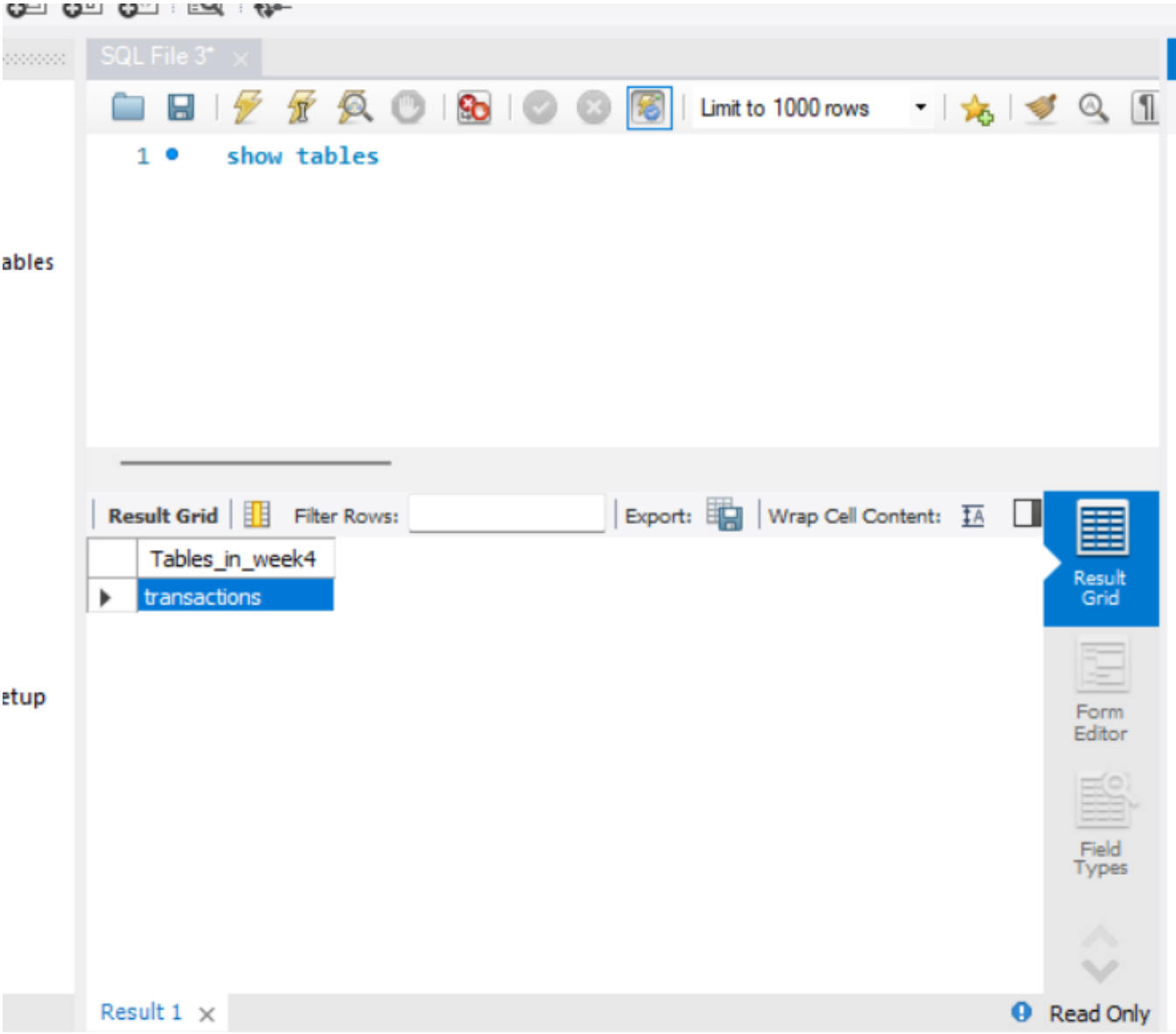
Assignment 1. RDS Resilience testing on AWS

Difficulty: ★★★☆☆.

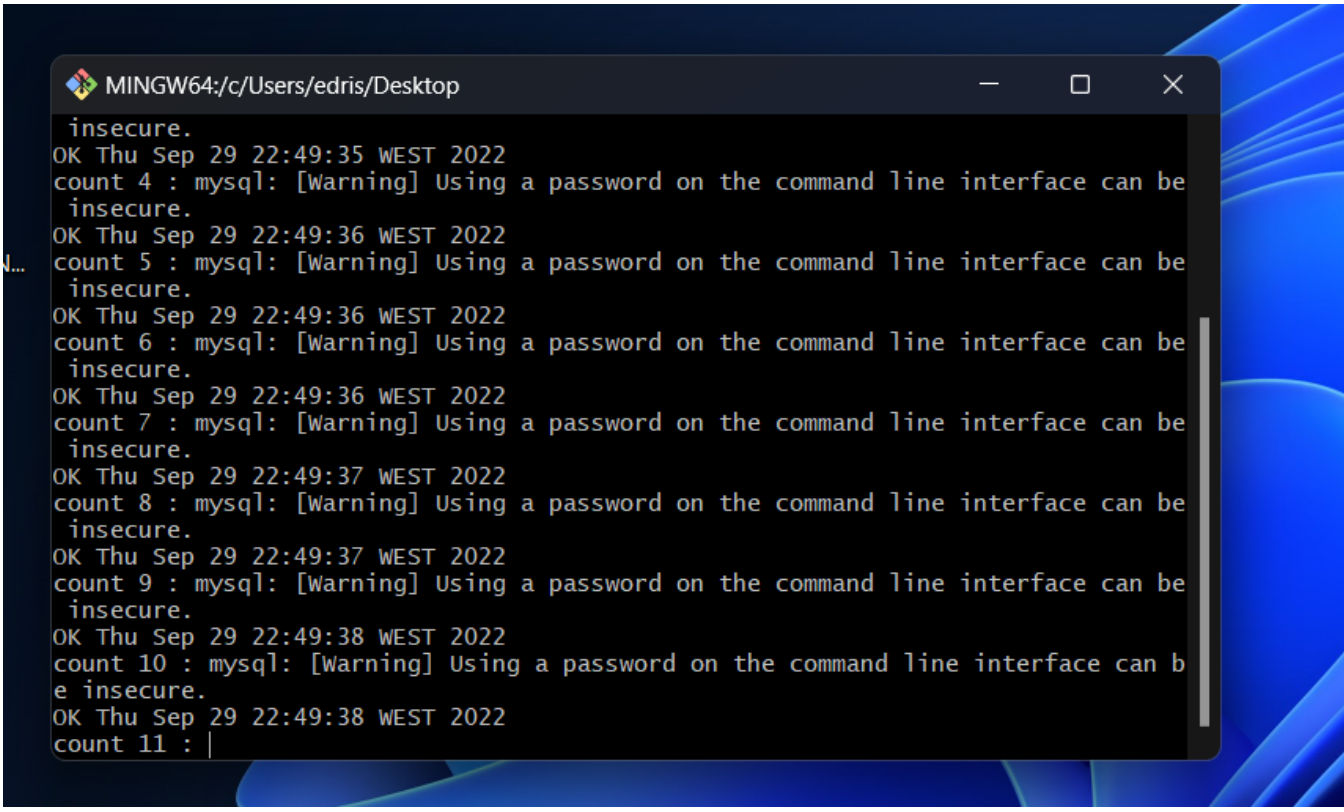
AWS documentation states that failover usually lasts between 60-120 seconds. The purpose of this assignment is to check if these estimations are correct, at least for MySQL databases.

1. Spin up a MySQL RDS instance, **with Muti-AZ enabled**
    a. Choose *Dev/Test* option and a *burstable* class db.t3.micro
    b. Set public access to Yes
    c. In "*Additional configuration*" specify an *Initial database name* to be created directly after instance is created. Name it as you wish, for instance, week8db.
2. Use SQL script to create a table in the database week8db. If you used different for the database, you'll need to change it in the script.



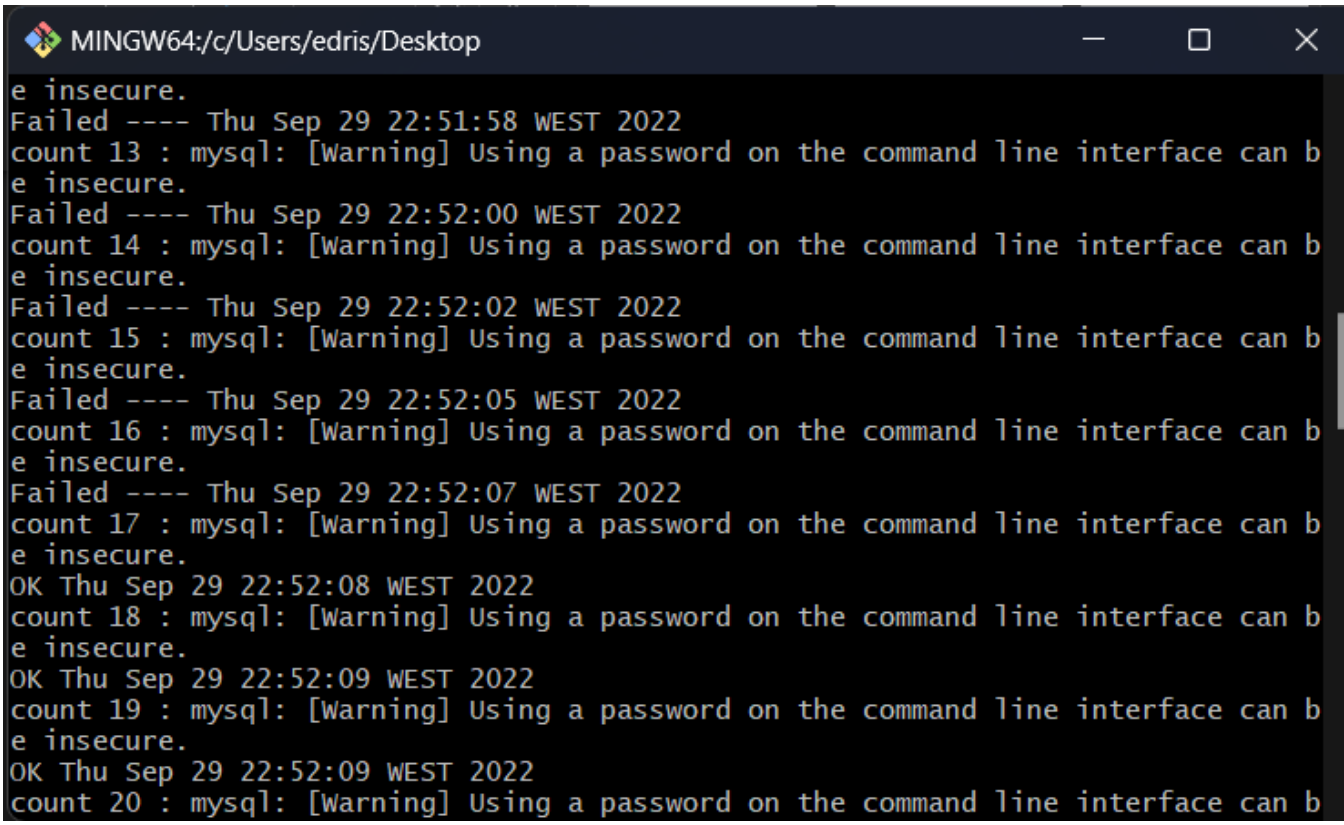
3. Amend this script so that it can connect to your database. It may require mysql command line client installation.
4. Execute the script . It will generate the output similar to the following:

```
count 2028 : OK Thu Oct 21 10:16:46 UTC 2021
count 2029 : OK Thu Oct 21 10:16:46 UTC 2021
count 2030 : OK Thu Oct 21 10:16:47 UTC 2021
count 2031 : Failed ---- Thu Oct 21 10:16:48 UTC 2021
count 2032 : Failed ---- Thu Oct 21 10:16:50 UTC 2021
count 2033 : Failed ---- Thu Oct 21 10:16:52 UTC 2021
count 2034 : Failed ---- Thu Oct 21 10:16:54 UTC 2021
count 2035 : Failed ---- Thu Oct 21 10:16:56 UTC 2021
count 2036 : Failed ---- Thu Oct 21 10:16:58 UTC 2021
```



5. Reboot your RDS instance, don’t forget to **check the failover option**!

6. Observe the output of the script. Terminate it once the connection is restored, make a screenshot and add it to your report. **How much time did the failover take in your case?**

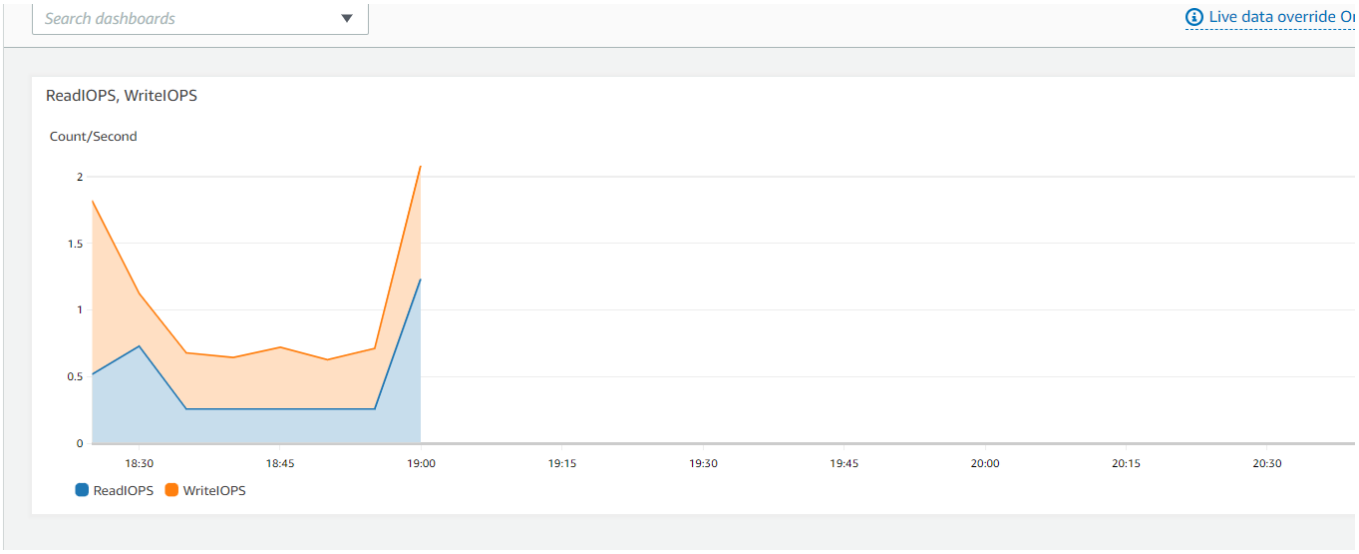


It only took a few seconds, 45 seconds.

7. In CloudTrail find the related events and try explaining them. Attach the screenshot to the report**. How do we know this failover was caused by the user initiated reboot?**

It tells use the IAM username of the user who caused the reboot. In this case: Edris. Else it would've been empty

8. Create a dashboard in CloudWatch to monitor read and write IOPS for your database. Amend the following script to match your configuration and execute it. How the workload is reflected in your dashboard? Do you have enough IOPS to sustain this load? Provide a couple of screenshots with the explanation what was going on at the database level.



My IOPS skyrocketed from nothing. I don’t have enough IOPS to sustain this load. If we sum up both the read and write IOPS we are constantly above 2 meaning we are under provisioned.

The database is writing faster than it can read.