Laboratory Practice Report

Practice 5

October 04, 2023

Computer Systems Engineering

Cloud Architecture

Prof. M.S. Rodolfo Luthe Ríos

Marco Ricardo Cordero Hernández

is727272@iteso.mx

ITESO

Universidad Jesuita
de Guadalajara

Abstract

Following the continuous learning path of cloud services and technology implementation, one of the cornerstones of any kind of modern application will be revised through the development of this practice: databases.

The main objective of this demonstration it’s to show how can users plan, configure and deploy relational databases through a cloud provider, along with the usual budget estimating. This is just one of the several steps needed for a real world application, however, this particular backbone would ultimately decide the workflow inside common developments for things such as high concurrency tools, administrative systems, or any kind of data intensive instrument that needs record storage in some way or another.

Another thing to note it’s that the contents of this document are not intended for database bounding demonstration, but rather initial creation and basic administration along with management through a cloud provider.

State of the Art

The modern world has undoubtedly change at its core functioning. The same old threats such as environmental dangers and world overpopulation are now accompanied by a new potential risk: data misuse. Nowadays, information industry has developed whole new societies and ways of living, something that previously could take even centuries to do so. The problem with this is that as easy as it’s to create new data entries for any type of thing imaginable, it also is to mishandle all these values.

Although data management isn’t something new, there are indeed new means of creation, administration, visualization and manipulation of it. Data is the new oil of the digital economy [1], and this fact poses new challenges for newcomers engineers and specialist. Nonetheless, although new technologies always posses some grade or another of risk, when they’re adopted to interact in the most seamless ways, they tend to evolve. Even so, data still needs to be stored and preserved, in order for it to be available for future mis/use. At its core basics, all transactions in this now data-driven world are supported by one versatile technology: databases.

Humans have always *needed* to preserve and share information, being through primitive paints, glyphs, books, or, in contemporary times, social media posts. But behind all these representations, data it’s stored in some form or another, and it’s the responsibility of the user of these data giving an appropriate interpretation. Beside that, the method of storing all this information is as important, and with the introduction of modern computers in the past century, a whole new world of opportunities was created.

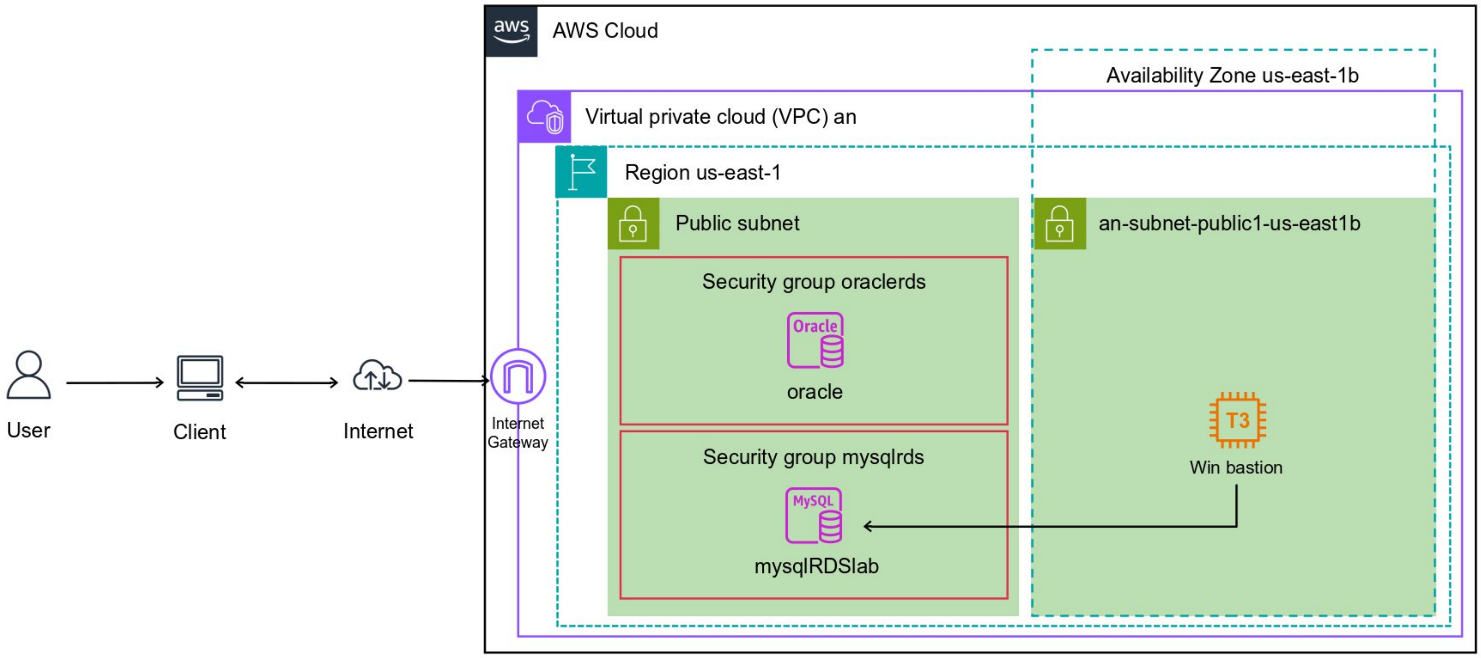
In the beginning, primitive databases were used for simple tasks such as record translation from physical writings to digital bytes representing the information that the former conveyed. It wasn’t until the 1970’s that Edgar F. Codd proposed the relational model, a database schema that disconnected information compendiums from physical information storage [2] [3], paving the road for what would become modern relational database management systems, better known as *RDBMS*.

Of course, non-relational systems would eventually become another way of representing data, however, classic structured schemes such as hospital records or banking histories are still backed up by this representation.

The goal of the current practice is to link two technologies to provide further demonstration of how technological pace it’s going extremely fast and how multiple information domains can interlink between them to provide greater support to existing developments. This of course it’s referring to AWS Relational Database Service [RDS] [4] inside the usual AWS Cloud. The service demonstration will be carried out by two relational databases instancing: Oracle [5] and MySQL [6].

Diagram

The following architecture it’s proposed as a graphic solution for the stated goals.

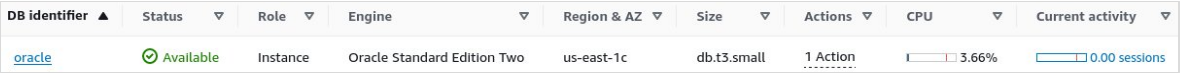


Practice Development

Oracle database

The previously mentioned service will be used as a first step to this development. RDS has its own dedicated panel inside the AWS console. Here, a new database will be created with the following features (**Note**: the account used for this practice allows Oracle usage, however, this database requires a license for its usage):

- Standard creation method (for custom tweaking)
- Oracle engine type (default; not custom)
- No multitenant architecture
- Edition: Standard Edition Two
- License: bring-your-own-license
- Version: Oracle 19 or newer
- Templates: Dev/Test
- Settings
    - Identifier: oracle
    - Credentials
        - Master username: admin
        - Master password: is727272-PASSWORD
- Instance class: burstable; db.t3.small (leave rest as default)
- Storage: 20 GB gp2 with storage autoscaling enabled
- Availability: do not create standby instance
- Connectivity
    - Don’t connect to EC2 resource
    - IPv4 network type
    - VPC: an-vpc (previously created) [this can’t be changed after creation]
    - Default subnet group
    - Allow public access
    - New VPC security group with *oraclerds* name
    - No preference for availability zone
    - Default certificate authority
    - Database port: 1521 (default)
- Authentication: password
- Default monitoring settings
- Additional configuration (leave defaults except for below listing)
    - Initial database name: orcl
    - Disable encryption



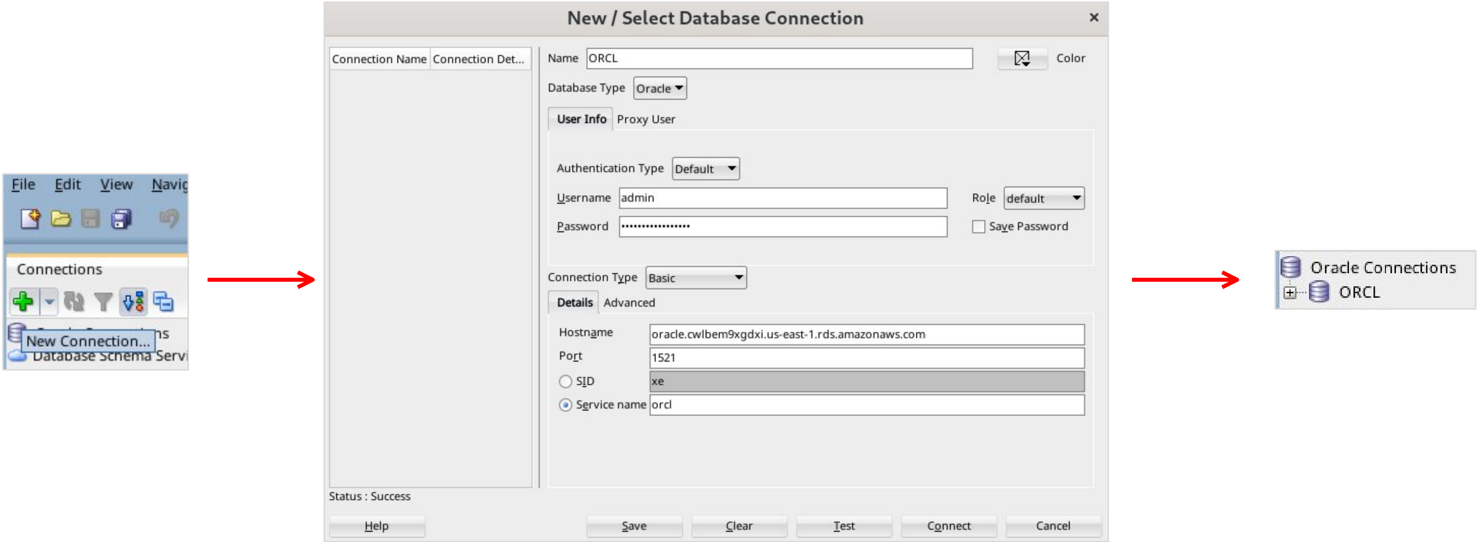
DB identifier ▲	Status ▽	Role ▽	Engine ▽	Region & AZ ▽	Size ▽	Actions ▽	CPU ▽	Current activity ▽
oracle	✓ Available	Instance	Oracle Standard Edition Two	us-east-1c	db.t3.small	1 Action	3.66%	0.00 sessions

Note: Before database initialization (which takes around 10 minutes), a suggested add-on window will be prompted; this can be ignored simply by closing it.

Note: For a supposed 100% utilization and single availability zone, the previous configurations will result in an approximate monthly charge of $28 USD.

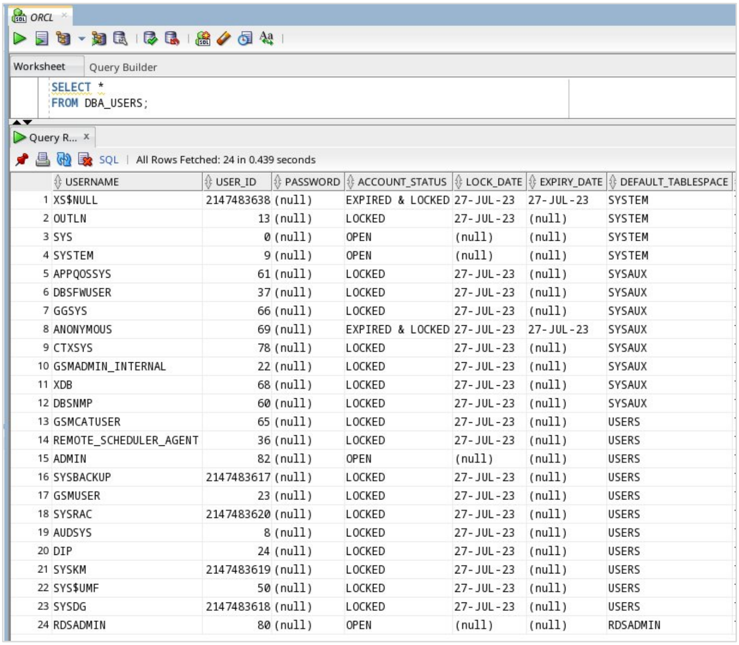
Several ways of connection to Oracle databases are available, however, SQL Developer tool [7] will be used in this practice. Once the database has been successfully created (at least in backing-up status) inside AWS, this application can be used to conduct the remote connection using the previously defined credentials. The connection endpoint can be found within the database details.

Inside SQL Developer GUI, a new connection has to be created. This can be found at the top left of the application. A window will be prompted, requesting connection information.



Name: ORCL
Database Type: Oracle
Authentication Type: Default
Username: admin
Role: default
Password: ••••••••••••••••
□ Save Password
Connection Type: Basic
Hostname: oracle.cwlbem9xgdxi.us-east-1.rds.amazonaws.com
Port: 1521
○ SID: xe
● Service name: orcl
Status : Success

As it can be seen, the connection status is successful. This can be seen once all data has been entered and then clicking on the "Test" button. Once this has been checked, the connection can be established and it'll be listed as an Oracle connection. Normal database transactions can be conducted once inside it, however, a simple user listing will suffice the purpose of this practice:



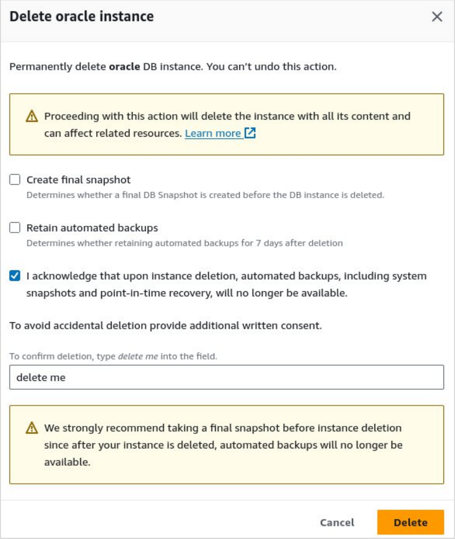
```
SELECT *
FROM DBA_USERS;
```

SQL | All Rows Fetched: 24 in 0.439 seconds

	USERNAME	USER_ID	PASSWORD	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DATE	DEFAULT_TABLESPACE
1	XS$NULL	2147483638	(null)	EXPIRED & LOCKED	27-JUL-23	27-JUL-23	SYSTEM
2	OUTLN	13	(null)	LOCKED	27-JUL-23	(null)	SYSTEM
3	SYS	0	(null)	OPEN	(null)	(null)	SYSTEM
4	SYSTEM	9	(null)	OPEN	(null)	(null)	SYSTEM
5	APPQOSSYS	61	(null)	LOCKED	27-JUL-23	(null)	SYSAUX
6	DBSFWUSER	37	(null)	LOCKED	27-JUL-23	(null)	SYSAUX
7	GGSYS	66	(null)	LOCKED	27-JUL-23	(null)	SYSAUX
8	ANONYMOUS	69	(null)	EXPIRED & LOCKED	27-JUL-23	27-JUL-23	SYSAUX
9	CTXSYS	78	(null)	LOCKED	27-JUL-23	(null)	SYSAUX
10	GSMADMIN_INTERNAL	22	(null)	LOCKED	27-JUL-23	(null)	SYSAUX
11	XDB	68	(null)	LOCKED	27-JUL-23	(null)	SYSAUX
12	DBSNMP	60	(null)	LOCKED	27-JUL-23	(null)	SYSAUX
13	GSMCATUSER	65	(null)	LOCKED	27-JUL-23	(null)	USERS
14	REMOTE_SCHEDULER_AGENT	36	(null)	LOCKED	27-JUL-23	(null)	USERS
15	ADMIN	82	(null)	OPEN	(null)	(null)	USERS
16	SYSBACKUP	2147483617	(null)	LOCKED	27-JUL-23	(null)	USERS
17	GSMUSER	23	(null)	LOCKED	27-JUL-23	(null)	USERS
18	SYSRAC	2147483620	(null)	LOCKED	27-JUL-23	(null)	USERS
19	AUDSYS	8	(null)	LOCKED	27-JUL-23	(null)	USERS
20	DIP	24	(null)	LOCKED	27-JUL-23	(null)	USERS
21	SYSKM	2147483619	(null)	LOCKED	27-JUL-23	(null)	USERS
22	SYS$UMF	50	(null)	LOCKED	27-JUL-23	(null)	USERS
23	SYSDG	2147483618	(null)	LOCKED	27-JUL-23	(null)	USERS
24	RDSADMIN	80	(null)	OPEN	(null)	(null)	RDSADMIN

MySQL database

Before this new database creation, the previous Oracle database has to be terminated due to account limitations.



Delete oracle instance

Permanently delete **oracle** DB instance. You can't undo this action.

⚠ Proceeding with this action will delete the instance with all its content and can affect related resources. Learn more

☐ Create final snapshot
Determines whether a final DB Snapshot is created before the DB instance is deleted.

☐ Retain automated backups
Determines whether retaining automated backups for 7 days after deletion

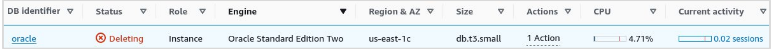
☑ I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.

To avoid accidental deletion provide additional written consent.

To confirm deletion, type *delete me* into the field.

delete me

⚠ We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available.

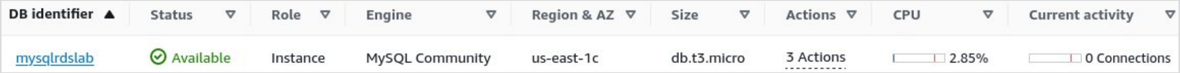


DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity
oracle	Deleting	Instance	Oracle Standard Edition Two	us-east-1c	db.t3.small	1 Action	4.71%	0.02 sessions

As soon as AWS starts the deletion process, the new database can be created. It should have the following features:

- Standard creation method (for custom tweaking)
- MySQL engine type
- Version: MySQL 8.x.x or newer
- Templates: Dev/Test
- Availability: Single DB Instance
- Settings
    - Identifier: mysqlRDSlab
    - Credentials
        - Master username: admin
        - Master password: is727272-PASSWORD
- Instance class: burstable; db.t3.micro
- Storage: 20 GB gp3 without autoscaling
- Connectivity
    - Don’t connect to EC2 resource
    - IPv4 network type
    - VPC: an-vpc (previously created) [this can’t be changed after creation]
    - Default subnet group
    - Allow public access
    - New VPC security group with *mysqlrds* name
    - No preference for availability zone
    - Default certificate authority
    - Database port: 3306 (default)
- Authentication: password
- Disable monitoring

- Additional configuration (leave defaults except for below listing)
    - Initial database name: mydb
    - Disable encryption

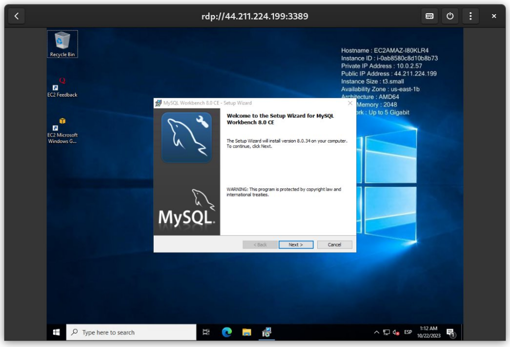


DB identifier ▲	Status ▽	Role ▽	Engine ▽	Region & AZ ▽	Size ▽	Actions ▽	CPU ▽	Current activity ▽
mysqlrdslab	✓ Available	Instance	MySQL Community	us-east-1c	db.t3.micro	3 Actions	2.85%	0 Connections

Note: As well as with the Oracle database, before MySQL database initialization (which also takes around 10 minutes), another suggested add-on window will be prompted; this can be ignored simply by closing it.

Note: For a supposed 100% utilization and single availability zone, the previous configurations will result in an approximate monthly charge of $37 USD (substantial difference with the Oracle RDS caused by gp3 storage type). It's also noted that this estimate has been made with a db.t3.small instance type, as db.t3.micro isn't available for budget estimation.

With the previously created Windows bastion EC2 [8] instance, MySQL access will be conducted through MySQL Workbench [9].



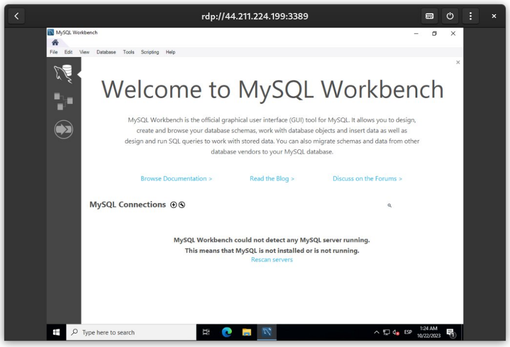
Hostname : EC2AMAZ-I80KLR4
Instance ID : i-0ab8580c8d10b8b73
Private IP Address : 10.0.2.57
Public IP Address : 44.211.224.199
Instance Size : t3.small
Availability Zone : us-east-1b
Architecture : AMD64
Memory : 2048
Network : Up to 5 Gigabit

Welcome to the Setup Wizard for MySQL Workbench 8.0 CE

The Setup Wizard will install version 8.0.34 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

rdp://44.211.224.199:3389
Recycle Bin
EC2 Feedback
EC2 Microsoft Windows G...
MySQL Workbench 8.0 CE - Setup Wizard
MySQL
< Back
Next >
Cancel
Type here to search
ESP
1:12 AM
10/22/2023



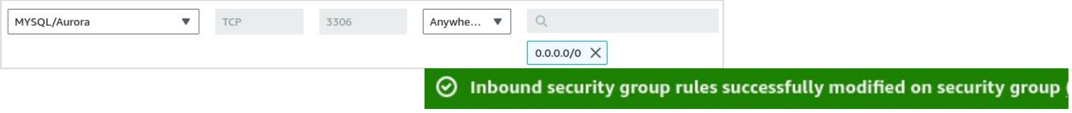
Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

MySQL Connections

MySQL Workbench could not detect any MySQL server running.
This means that MySQL is not installed or is not running.

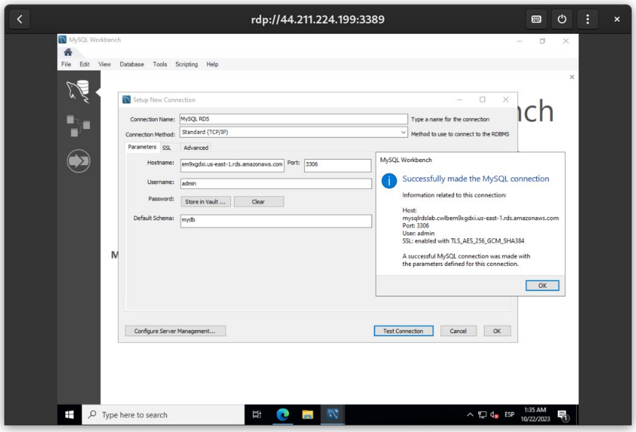
Before making the connection, the database's security group has to be modified to accept TCP connections at port 3306 from the bastion's security group.



MYSQL/Aurora ▼ | TCP | 3306 | Anywhe... ▼ | 0.0.0.0/0 ✕

✓ Inbound security group rules successfully modified on security group

Once this step it's done, connection can be established.



Setup New Connection

Connection Name: MySQL RDS — Type a name for the connection
Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: em9xgdxi.us-east-1.rds.amazonaws.com Port: 3306
Username: admin
Password: Store in Vault ... Clear
Default Schema: mydb

MySQL Workbench

Successfully made the MySQL connection

Information related to this connection:

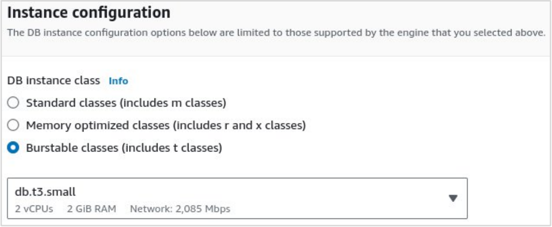
Host:
mysqlrdslab.cwlbem9xgdxi.us-east-1.rds.amazonaws.com
Port: 3306
User: admin
SSL: enabled with TLS_AES_256_GCM_SHA384

A successful MySQL connection was made with the parameters defined for this connection.

MySQL database modification

Just to demonstrate how changes can be made inside RDS, MySQL database will be modified:

- Instance type: db.t3.micro —> db.t3.small
- Multi-AZ deployment
- Storage: 20GB —> 30GB



Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

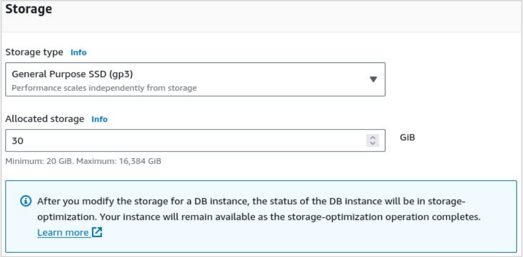
DB instance class Info

○ Standard classes (includes m classes)

○ Memory optimized classes (includes r and x classes)

● Burstable classes (includes t classes)

db.t3.small
2 vCPUs 2 GiB RAM Network: 2,085 Mbps



Storage

Storage type Info

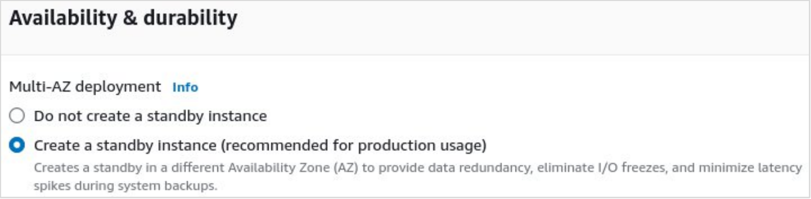
General Purpose SSD (gp3)
Performance scales independently from storage

Allocated storage Info

30 GiB

Minimum: 20 GiB. Maximum: 16,384 GiB

After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes. Learn more



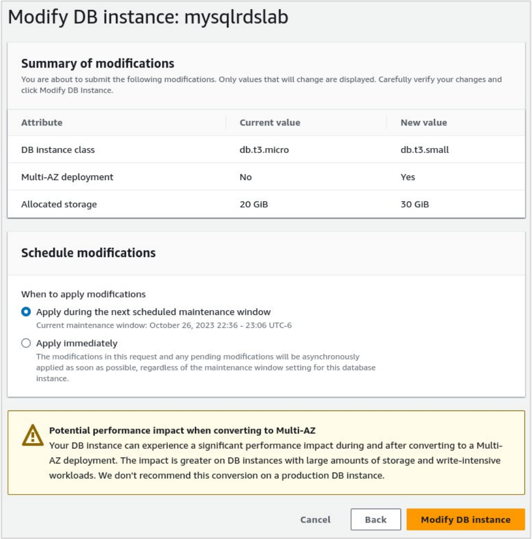
Availability & durability

Multi-AZ deployment Info

○ Do not create a standby instance

● Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

Before changes are rolled over, a non-blocking warning message inside made changes summary will indicate to the user that multi-AZ conversion will potentially affect and reduce database performance.



Modify DB instance: mysqlrdslab

Summary of modifications

You are about to submit the following modifications. Only values that will change are displayed. Carefully verify your changes and click Modify DB Instance.

Attribute	Current value	New value
DB instance class	db.t3.micro	db.t3.small
Multi-AZ deployment	No	Yes
Allocated storage	20 GiB	30 GiB

Schedule modifications

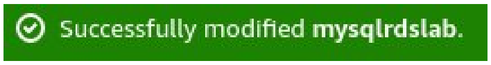
When to apply modifications

● Apply during the next scheduled maintenance window
Current maintenance window: October 26, 2023 22:36 - 23:06 UTC-6

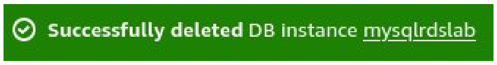
○ Apply immediately
The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

Potential performance impact when converting to Multi-AZ
Your DB instance can experience a significant performance impact during and after converting to a Multi-AZ deployment. The impact is greater on DB instances with large amounts of storage and write-intensive workloads. We don't recommend this conversion on a production DB instance.

Inside this section, the user can also define whether to apply changes immediately or during next maintenance period (default). After confirmation, a successful prompt will be emitted; after these modifications are made, the database can be terminated.



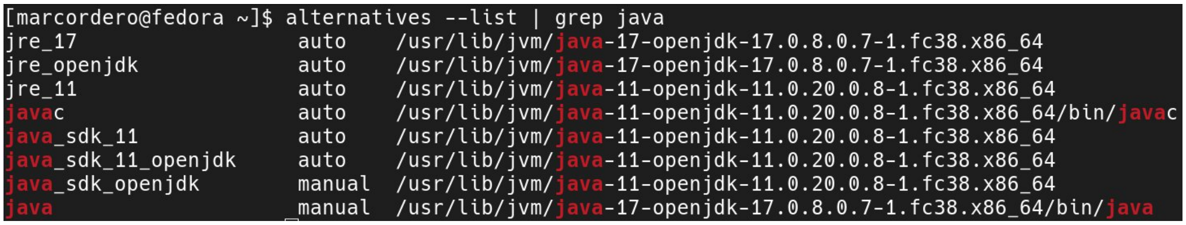
Successfully modified **mysqlrdslab.**



Successfully deleted DB instance mysqlrdslab

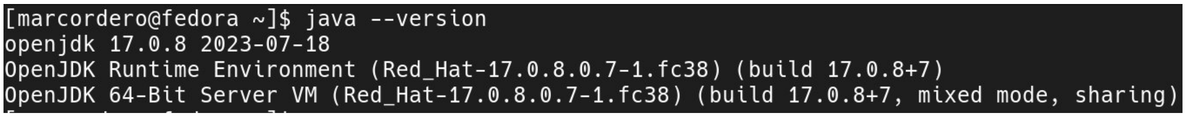
Problems and Solutions

This time, a single problem arose within Oracle section, specifically with SQL Developer. This series of practices have all been developed through a Fedora Workstation [10] Linux distribution. When using an operative system so nonrestrictive such as Fedora or any other kind of Linux based system, dependencies and installations can quickly become a mess. In this case, the system used had two different versions of Java installed in it: 11 and 17.



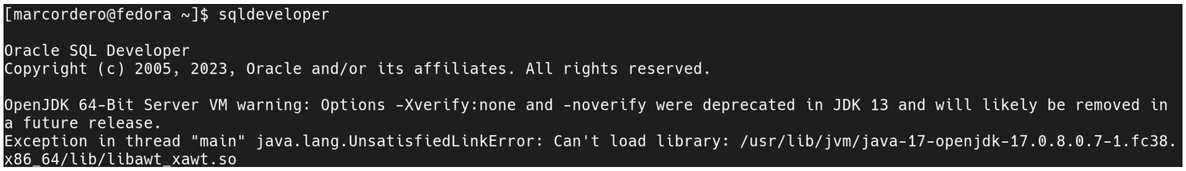
```
[marcordero@fedora ~]$ alternatives --list | grep java
jre_17                  auto     /usr/lib/jvm/java-17-openjdk-17.0.8.0.7-1.fc38.x86_64
jre_openjdk             auto     /usr/lib/jvm/java-17-openjdk-17.0.8.0.7-1.fc38.x86_64
jre_11                  auto     /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.fc38.x86_64
javac                   auto     /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.fc38.x86_64/bin/javac
java_sdk_11             auto     /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.fc38.x86_64
java_sdk_11_openjdk     auto     /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.fc38.x86_64
java_sdk_openjdk        manual   /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.fc38.x86_64
java                    manual   /usr/lib/jvm/java-17-openjdk-17.0.8.0.7-1.fc38.x86_64/bin/java
```

SQL Developer download page indicates that the tool uses JDK 11. This machine had version 17 configured as default.



```
[marcordero@fedora ~]$ java --version
openjdk 17.0.8 2023-07-18
OpenJDK Runtime Environment (Red_Hat-17.0.8.0.7-1.fc38) (build 17.0.8+7)
OpenJDK 64-Bit Server VM (Red_Hat-17.0.8.0.7-1.fc38) (build 17.0.8+7, mixed mode, sharing)
```

Tool execution with this version will result in the following error:

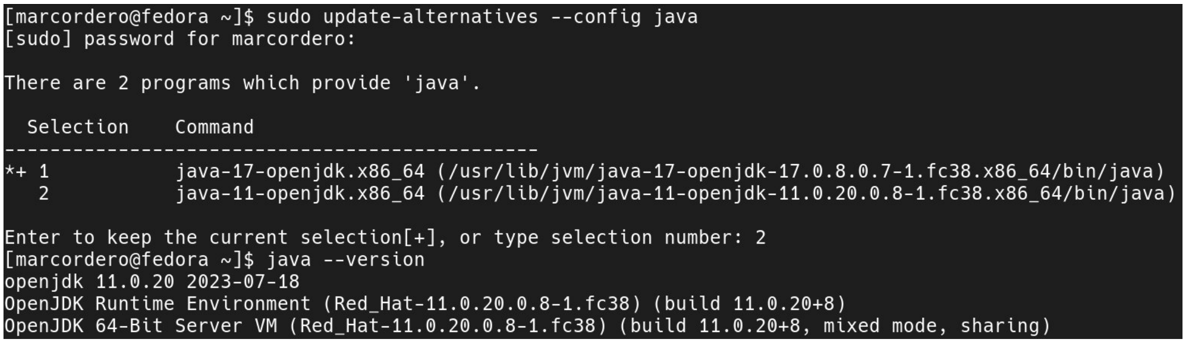


```
[marcordero@fedora ~]$ sqldeveloper

Oracle SQL Developer
Copyright (c) 2005, 2023, Oracle and/or its affiliates. All rights reserved.

OpenJDK 64-Bit Server VM warning: Options -Xverify:none and -noverify were deprecated in JDK 13 and will likely be removed in
a future release.
Exception in thread "main" java.lang.UnsatisfiedLinkError: Can't load library: /usr/lib/jvm/java-17-openjdk-17.0.8.0.7-1.fc38.
x86_64/lib/libawt_xawt.so
```

To fix this, a simple version change has to be made; Although it's an effective resolution method, when undone, it can lead to unexpected behavior in other areas of the system (solution to this is beyond the scope of this work). After this change, the tool can be executed normally.

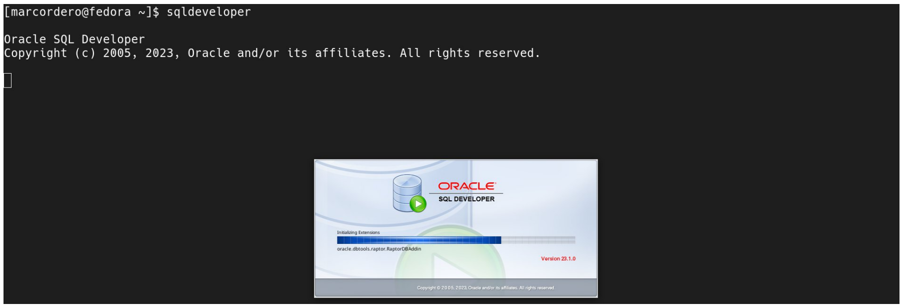


```
[marcordero@fedora ~]$ sudo update-alternatives --config java
[sudo] password for marcordero:

There are 2 programs which provide 'java'.

  Selection    Command
-----------------------------------------------
*+ 1           java-17-openjdk.x86_64 (/usr/lib/jvm/java-17-openjdk-17.0.8.0.7-1.fc38.x86_64/bin/java)
   2           java-11-openjdk.x86_64 (/usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.fc38.x86_64/bin/java)

Enter to keep the current selection[+], or type selection number: 2
[marcordero@fedora ~]$ java --version
openjdk 11.0.20 2023-07-18
OpenJDK Runtime Environment (Red_Hat-11.0.20.0.8-1.fc38) (build 11.0.20+8)
OpenJDK 64-Bit Server VM (Red_Hat-11.0.20.0.8-1.fc38) (build 11.0.20+8, mixed mode, sharing)
```



```
[marcordero@fedora ~]$ sqldeveloper

Oracle SQL Developer
Copyright (c) 2005, 2023, Oracle and/or its affiliates. All rights reserved.
```

ORACLE
SQL DEVELOPER
Initializing Extensions
oracle.dbtools.raptor.RaptorDBAddin
Version 23.1.0
Copyright © 2005, 2023, Oracle and/or its affiliates. All rights reserved.

Experiments and Results

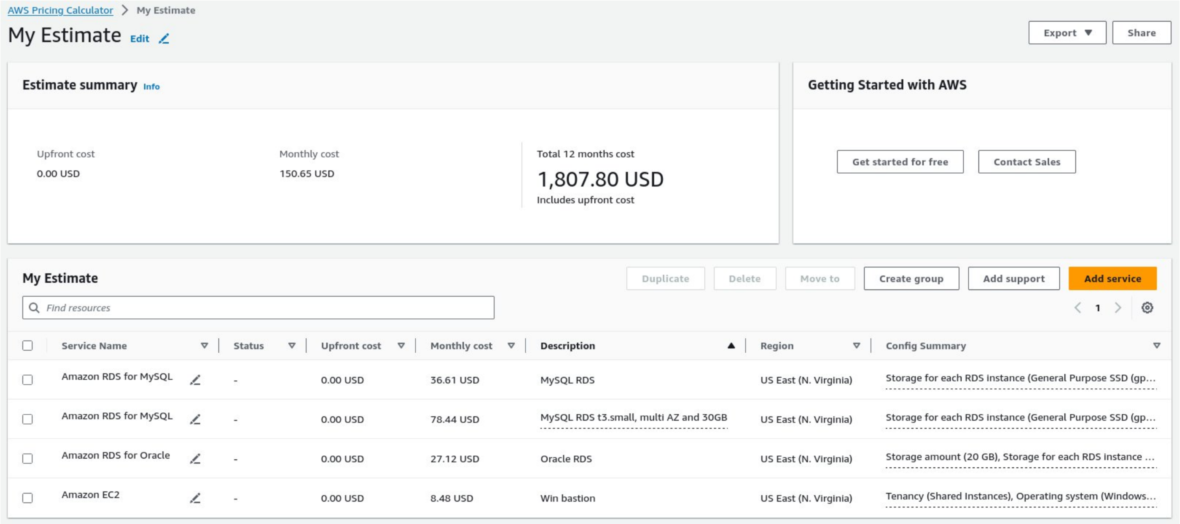
As this practice had a lot of straightforward sections and well defined steps, no experiments were conducted this time, however, a real application architecture implementation in which cloud usage it's put to practice would surely be interesting.

Budget Justification

This development had four major components:

- 1 EC2 bastion instance
- 1 Oracle RDS
- 2 MySQL RDS configurations
    - 1 with 20 GB of storage, db.t3.micro instance type, single availability zone
    - 1 with 30 GB of storage, db.t3.small instance type, multiple availability zones

Taking into consideration these services, the budget for this implementation would be the following:



My Estimate

Estimate summary

Upfront cost
0.00 USD

Monthly cost
150.65 USD

Total 12 months cost
1,807.80 USD
Includes upfront cost

My Estimate

☐	Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary
☐	Amazon RDS for MySQL	-	0.00 USD	36.61 USD	MySQL RDS	US East (N. Virginia)	Storage for each RDS instance (General Purpose SSD (gp...
☐	Amazon RDS for MySQL	-	0.00 USD	78.44 USD	MySQL RDS t3.small, multi AZ and 30GB	US East (N. Virginia)	Storage for each RDS instance (General Purpose SSD (gp...
☐	Amazon RDS for Oracle	-	0.00 USD	27.12 USD	Oracle RDS	US East (N. Virginia)	Storage amount (20 GB), Storage for each RDS instance ...
☐	Amazon EC2	-	0.00 USD	8.48 USD	Win bastion	US East (N. Virginia)	Tenancy (Shared Instances), Operating system (Windows...

Conclusions

Having had a very technical career and several opportunities of using database management systems, it's such a delight implementing the core data support for possible application or other data driven developments. It's such an amazing feat combining cloud usage and data storage to create a seamless solution without having to deal with overcomplicated resource management such as computer equipment or update provision.

As stated in the introductory sections, maybe this implementation isn't that meaningful as data creation and manipulation transactions weren't conducted, however, what's been done it's an important and crucial step in any application creation.

Bibliography

[1] J. Toonders. 'Data Is the New Oil of the Digital Economy'. [Online]. Available: https://www.wired.com/insights/2014/07/data-new-oil-digital-economy/.

[2] Quickbase. ' A Timeline of Database History & Database Management '. [Online]. Available: https://www.quickbase.com/articles/timeline-of-database-history.

[3] IBM. 'Edgar F. Codd'. [Online]. Available: https://www.ibm.com/ibm/history/exhibits/builders/builders_codd.html.

[4] Aws.amazon.com. ' Amazon Relational Database Service'. [Online]. Available: https://aws.amazon.com/rds/.

[5] Oracle. 'Database'. [Online]. Available: https://www.oracle.com/database/.

[6] MySQL. 'MySQL Standard Edition'. [Online]. Available: https://www.mysql.com/products/standard/.

[7] Oracle. 'SQL Developer'. [Online]. Available: https://www.oracle.com/database/sqldeveloper/.

[8] Aws.amazon.com. 'Amazon EC2'. [Online]. Available: https://aws.amazon.com/es/ec2/.

[9] MySQL. 'MySQL Workbench'. [Online]. Available: https://www.mysql.com/products/workbench/.

[10] The Fedora Project. 'Fedora Workstation'. [Online]. Available: https://fedoraproject.org/workstation/.