RDS Migration- Blue/Green Deployment.

1. Knowledge Base

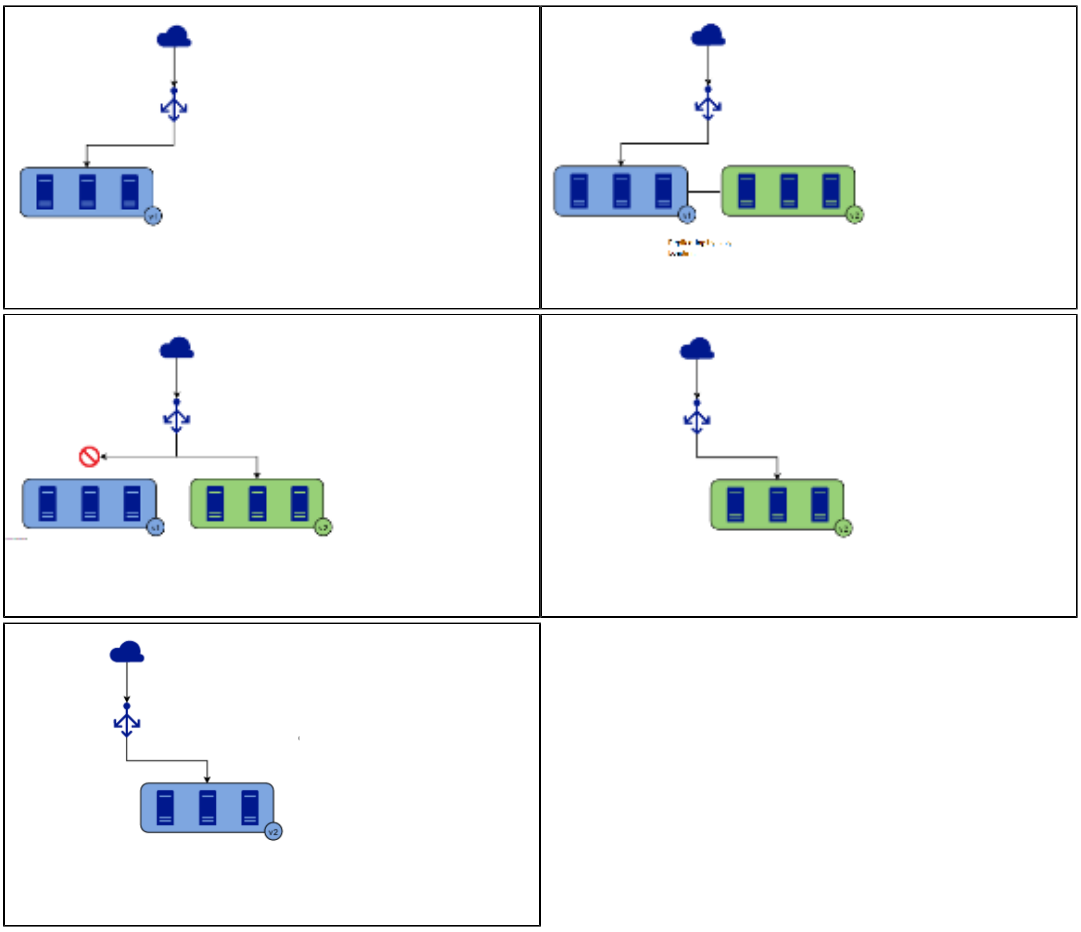
1.1. **Blue/Green Deployment**

Step 1: Blue env to green Step 2: B/G deployment Step 3: Switch over started Step 4: App connected Step 5: B/G deployment success

Replicating data to green by lazy loading **Downtime expecting(2 mins)**

Depending on how much replica is pending from blue to green.

The above estimation we got from the pilot-service Application



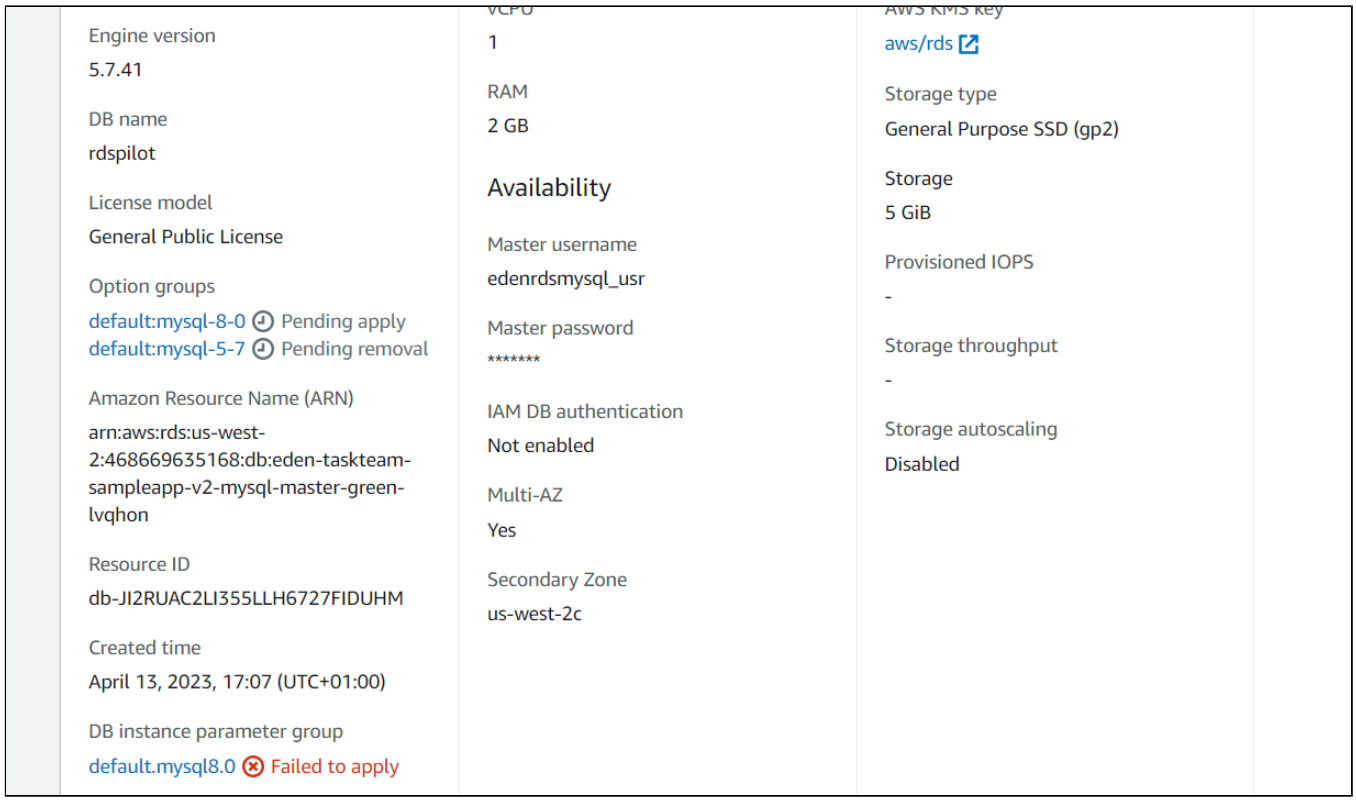
Before starting the b/g deployment, one should be aware of some facts

1.1.1. Blue/Green Environment Creation

- B/G Env. Creation has 3 steps, which take approximately 45 to 50 mins,
  - Creating a read replica of the source
  - Db engine version upgrade
  - Configure backups

Note:

- Above observation of B/G creation time is tested with one table with 103,250 records.
- If any of the above steps fail, b/g deployment tries to fix the issue.



- Before upgrading the MySQL 8.0 version, it checks the prepatch compatibility.
  These are the checks it will carry before upgrading to 8

```
1) Usage of old temporal type
No issues found.

2) Usage of db objects with names conflicting with new reserved keywords
No issues found.

3) Usage of utf8mb3 charset
The following objects use the utf8mb3 character set. It is recommended to convert them to use utf8mb4 instead,
for improved Unicode support.
More Information:
https://dev.mysql.com/doc/refman/8.0/en/charset-unicode-utf8mb3.html

mysql - schema's default character set: utf8


4) Table names in the mysql schema conflicting with new tables in 8.0
No issues found.

5) Partitioned tables using engines with non native partitioning
No issues found.

6) Foreign key constraint names longer than 64 characters
No issues found.

7) Usage of obsolete MAXDB sql_mode flag
No issues found.

8) Usage of obsolete sql_mode flags
No issues found.

9) ENUM/SET column definitions containing elements longer than 255 characters
No issues found.
```

```
10) Usage of partitioned tables in shared tablespaces
No issues found.

11) Circular directory references in tablespace data file paths
No issues found.

12) Usage of removed functions
No issues found.

13) Usage of removed GROUP BY ASC/DESC syntax
No issues found.

14) Removed system variables for error logging to the system log configuration
No issues found.

15) Removed system variables
No issues found.

16) System variables with new default values
No issues found.

17) Schema inconsistencies resulting from file removal or corruption
No issues found.

18) Issues reported by 'check table x for upgrade' command
No issues found.

19) The definer column for mysql.events cannot be null or blank.
No issues found.

20) Tables with dangling FULLTEXT index reference
No issues found.

21) Routines with deprecated keywords in definition
No issues found.

22) DB instance must have enough free disk space
No issues found.

23) Creating indexes larger than 767 bytes on tables with redundant row format might cause the tables to be
inaccessible.
No issues found.

24) The tables with redundant row format can't have an index larger than 767 bytes.
No issues found.

25) Column definition mismatch between InnoDB Data Dictionary and actual table definition.
No issues found.
```

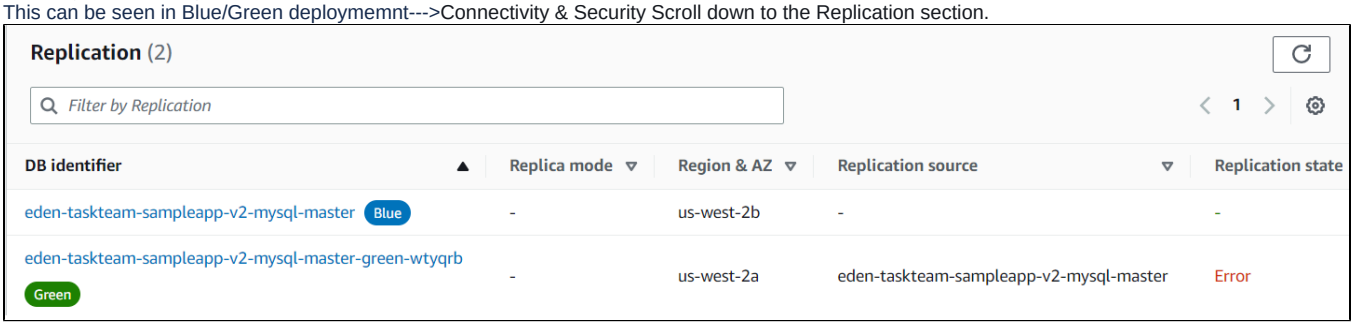
Once b/g env is created, the data in blue will be **asynchronously** replicating to green. But due to **network issues** or **i/o issues**, there may be a delay to replicate the data to green. This can be checked using the ReplicaLag metric in CloudWatch. (This is shown further below).

1.1.2. Switch Over

A switchover promotes the green environment to be the new production environment

1.1.2.1. Cancel to Switch over

If the replica stopped completely and shows an error, then restart the whole process again by deleting the blue/green deployment. Blue green deployment internally deletes the green environment.
This can be seen in Blue/Green deploymemnt--->Connectivity & Security Scroll down to the Replication section.



1.1.2.2. Proceed to Switch over

There are 2 factors

- ReplicaLag metrics of green env
- The maximum number of connections of DB instance in blue env. This can be checked from performance insight if it is enabled or from the DatabaseConnections metric in CloudWatch of blue env.

ReplicaLag should be near zero.

> 0- replica is continuing in the background

< 0 replica is not active

During the switchover, writes are cut off from databases in both environments. **During this cut-off time, if any operation takes place from the application, we are losing the data here.**

1.1.2.2.1. Choose the quiet time to switch over to production env

Check DatabaseConnections from CloudWatch for 1 month and decide the time based on low database connections.

1.1.2.3. Success Switch over

Check the pod. It should run without any interruption.

In the edge case, if the pod goes crash back loop status (pod is not able to connect to new db instance due to credential issue), re-deploy the CFN

1.1.2.4. FAQ on Switch over

2. Demo

2.1. Prerequisite

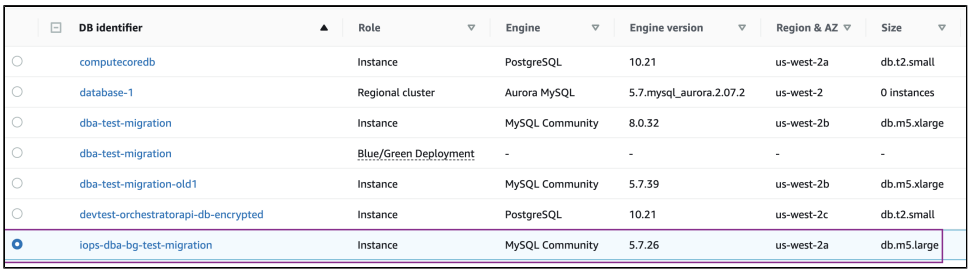
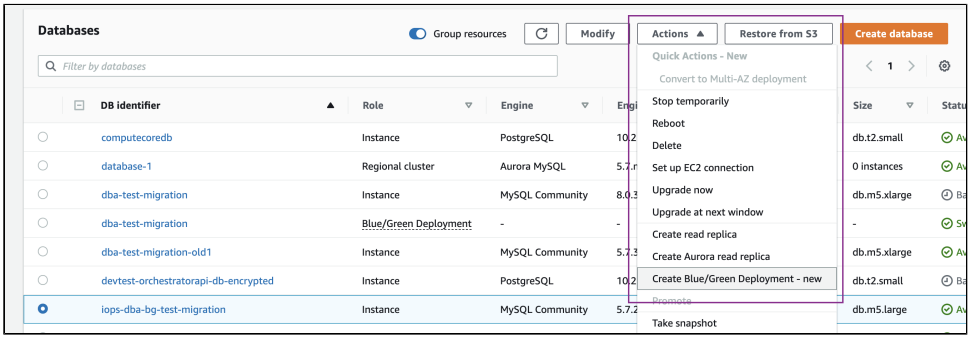
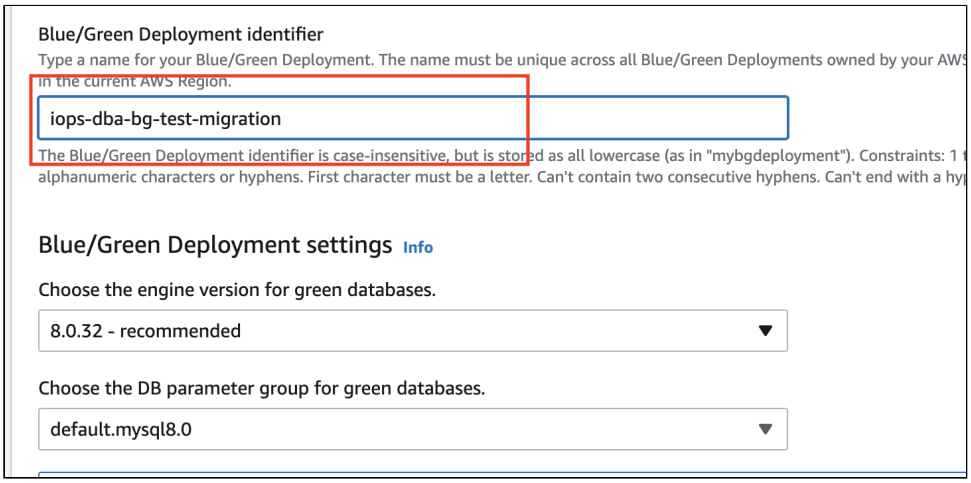
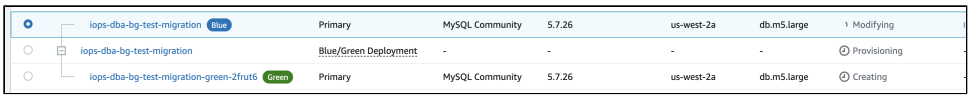
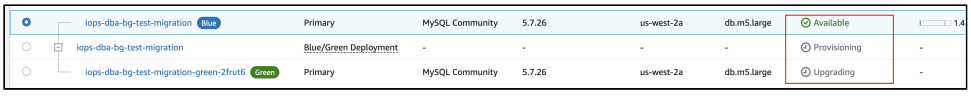
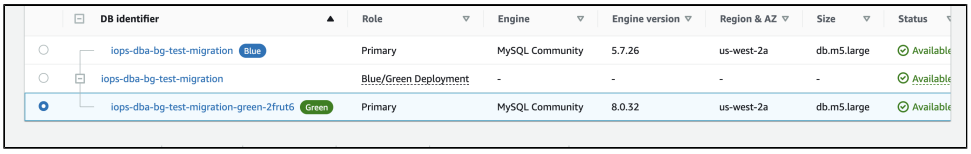
- Check the MySQL Connector version which connects to MySQL 8.0

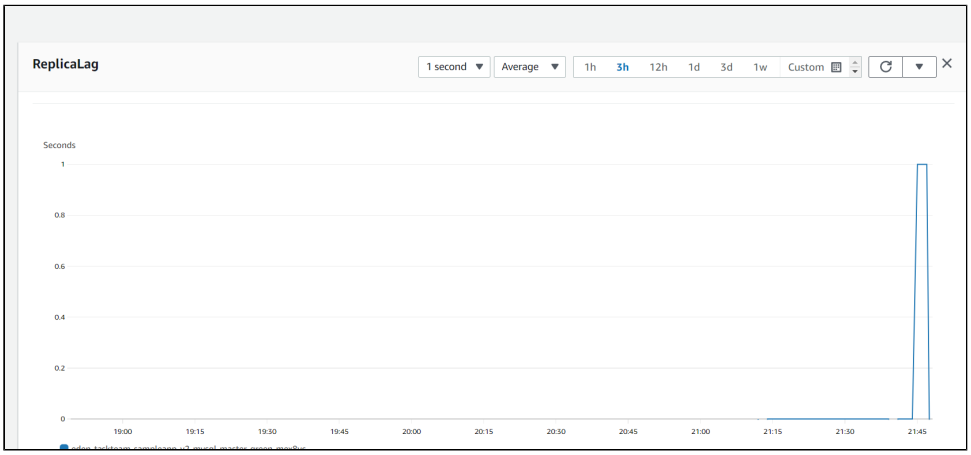
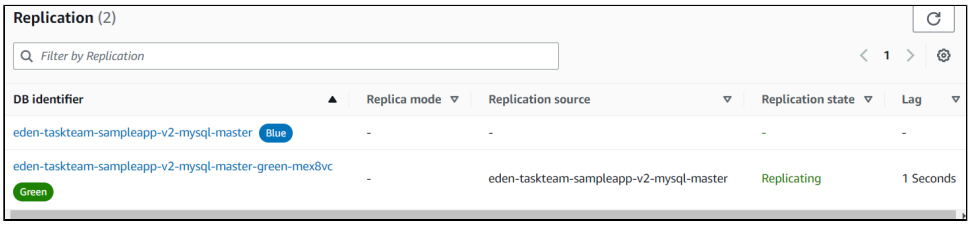
Note: mysql-connector-j is recommended over mysql-connector-java

2.2. Creating a blue/green deployment

One should have a DevOps role(If not, please create a service ticket: https://concur.service-now.com/csp?id=sc_cat_item&sys_id=da0398df1b3f6f0068ff0dc8cd4bcb6e)

	Description	Instructions	ScreenShot

1	Go to RDs services	Select the rds instance	
2	Go to the Actions dropdown	Choose to Create Blue /Green Deployment - new	
3	Enter relevant info in Create Blue /Deployment page	• Provide the Blue /Green Deployment identifier. • Choose a DB engine version if you want to test a DB engine version upgrade.(Here used 8.0.32) • Choose a DB parameter group to associate with the DB instances in the green environment.(Used default)	
4	The new instance provisioning starts	---	
5	The green instance upgrade starts	---	
6	The green instance will be available after upgrade		

7	At this stage, you can test your staging environment.	Connect to the green instance using vault credential and MySQL workbench.	
8	Monitor the Replica lag. There are 2 ways to monitor ReplicaLag metric 1. Using green instance 2. Using blue /green deployment	• Select the green DB identifier • Then select MonitoringCloudWatch (it may be on the next page). It should be close to zero while switching over to green, which reduces downtime.	
9	Monitor the Replica lag using b/g deployment	Select B/G deploymentConnectivity & Security Scroll down to the Replication section.	

2.3. Connecting to green db instance via MySQL workbench in Integration env

Refer:
Connect to RDS instance

Retrieve namespace secret from Vault

2.4. Switching a Blue/Green deployment

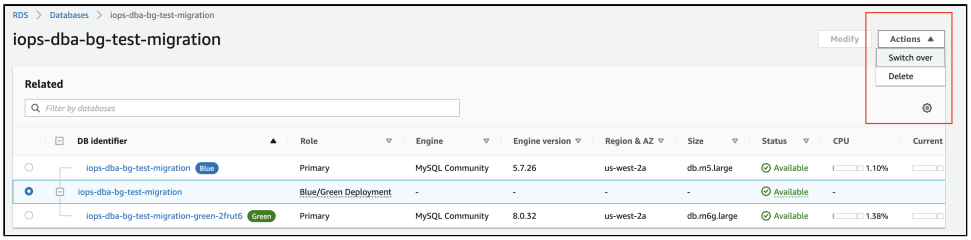
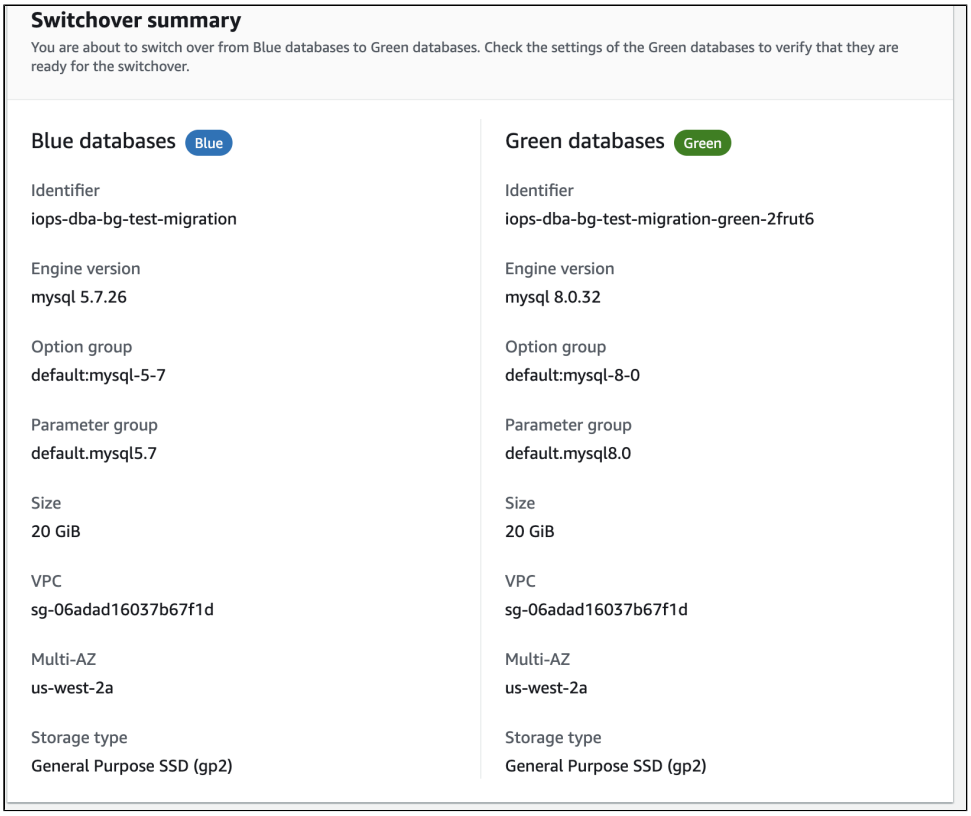
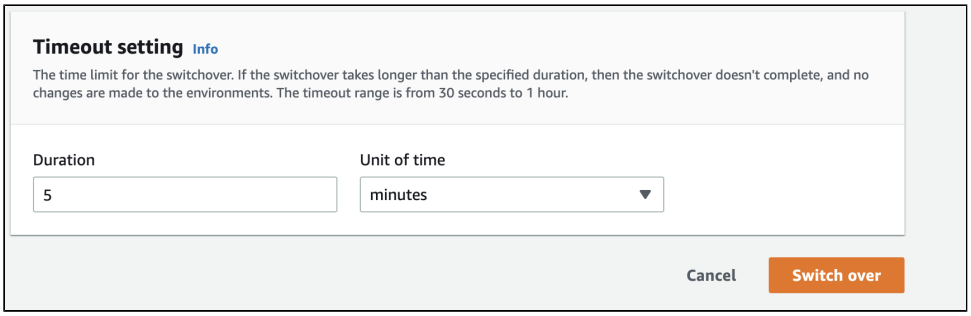
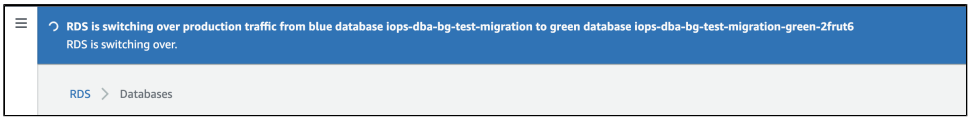
Note: **The switchover results in downtime. The downtime is usually under two minutes, but it can be longer depending on your workload.**

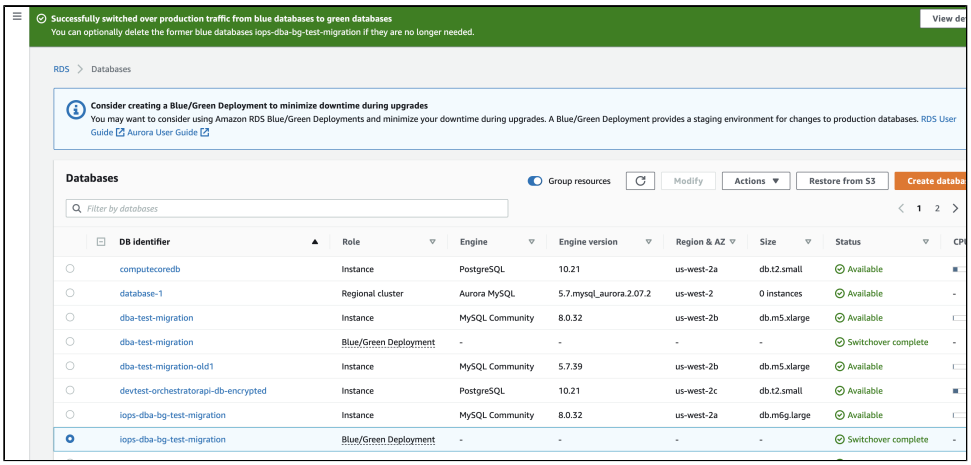
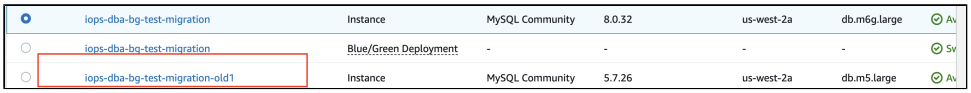
After the switchover is complete, the DB instances that were in the green environment become the new production DB instances. The names and endpoints in the current production environment are assigned to the newly promoted production environment.

Please read and follow the **Switchover best practices** from the below AWS document.

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments-switching.html

	Description	Instructions	Screenshot

1	Start the switch over	Select the b/g deployment radio button Select the Actions, then choose Switch over	
2	Review the Switch over summary	1. On the **Switchover** page, review the switchover summary. Make sure the resources in both environments match what you expect. If they don't, choose **Cancel**.	
3	Enter the timeout	For **Timeout**, enter the time limit for the switchover.	
4	Switch over is going to start	Apply Switch over	

5	Successful switch over	------	
6	Switchover completion	----	
7	Rename the Previous instance as <db-instance-name>-old1	-----	

2.5. Check Pods

Check the pod's status. Delete the pod and check pod started successfully or not. If pods not start successfully due to access denied to connect to DB instance, then re deploy the CFN.

2.6. Re deploy CloudFormation

Redeploy CFN for 2 reasons.

- If pods are not started successfully due to vault credential issues
- After the switch over, pods started successfully, but CFN is in drifted status.

Modify the following properties in CFN

- Family Properties to MySQL8.0
- EngineVersion to 8.0

```
    RDSParamGroup:
      Type: AWS::RDS::DBParameterGroup
      Properties:
-       Family: MySQL8.0
        Description: CloudFormation DB Parameter Group
        Parameters:
          log_bin_trust_function_creators: '1'
  @@ -438,7 +438,7 @@ Resources:
          - true
          - false
        Engine: MySQL
-       EngineVersion: '8.0'
        MasterUsername: !Ref DBUser
```

3. SwitchOver Tracking from various team

		Switch over time(in mins)			
Team	**Replica DB(Y/N)**	**Integration**	**US2**	**EU2**	**CCPS**
Budget					
Cards					
ClientAudit					
Curaçao					
DaVinci					
eps-mysql-master???					
Funds Management					
Invoice config					
Invoice core					
JPT	8.0				
perdiem-mysql-master	Y (8.0.32)	~1	~1	~1	0
Purchase(purgeorchestration-db-mysql-master)					
Quick expense					
Smart expense					
Incredibles					
spend-alpha-purchase-mysql-master??					
CRService					
spend-ems-rds-mysql-master??					
spend-ems-sandbox					
Wizard					

spend-pay-classic-service-mysql-master??					
spend-pay-global-service-mysql-master??					
spend-pay-invoice-ipm-service-mysql-master??					
spend-tango-invoicecapturedb-mysql-master??					
Request DB					
ua-benchmark-us2-mysql-master??					
us2-db-fictitiousmeal-mysql-master??					
us2-db-fictitiousmeal-mysql-master-new??					
us2-db-fictitiousmeal-mysql-master-old??					
vmsdb-mysql-master??					

4. Ref

4.1. AWS

This is the documentation created by DBAs based on the test migration done on a restored test database. Each team **must do test migrations** using their own data and application connectivity on MySQL8.

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments.html

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments-overview.html

4.2. MySQL

https://dev.mysql.com/doc/refman/8.0/en/upgrading-from-previous-series.html

5. Important note from AWS

- RDS B/G deployments are not supported via AWS CloudFormation [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments-overview.html#blue-green-deployments-limitations].
- There is no rollback feature for RDS B/G deployment currently available. You can review approach shared in blog post from @Matkar, Rajesh which you could leverage and customize https://aws.amazon.com/blogs/database/performing-major-version-upgrades-for-amazon-aurora-mysql-with-minimum-downtime/

AWS Document followed for the deployment: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments-creating.html

6. Rollback

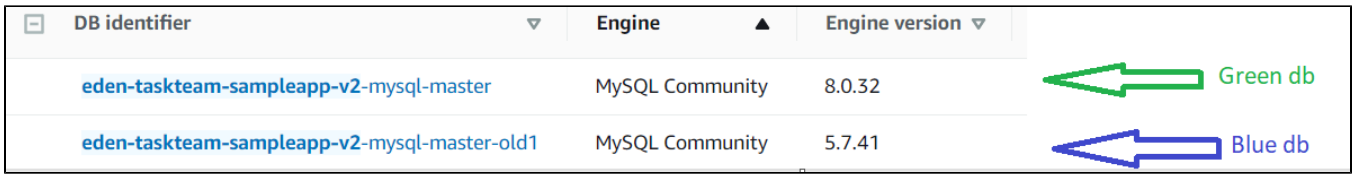
There is no rollback feature for RDS B/G deployment currently available.We are doing it manually. We can rollback to the previous version using renaming the db instances

6.1. Renaming the DB instance

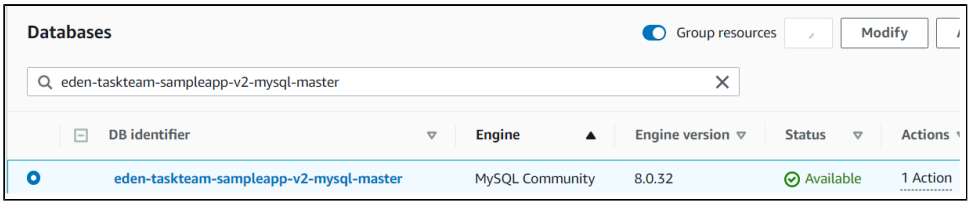
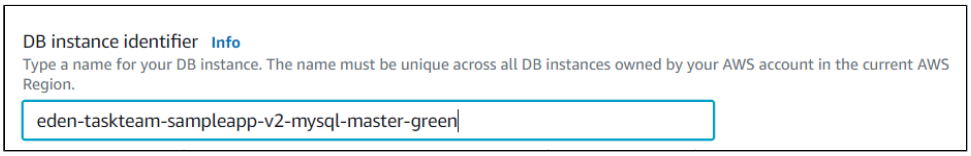
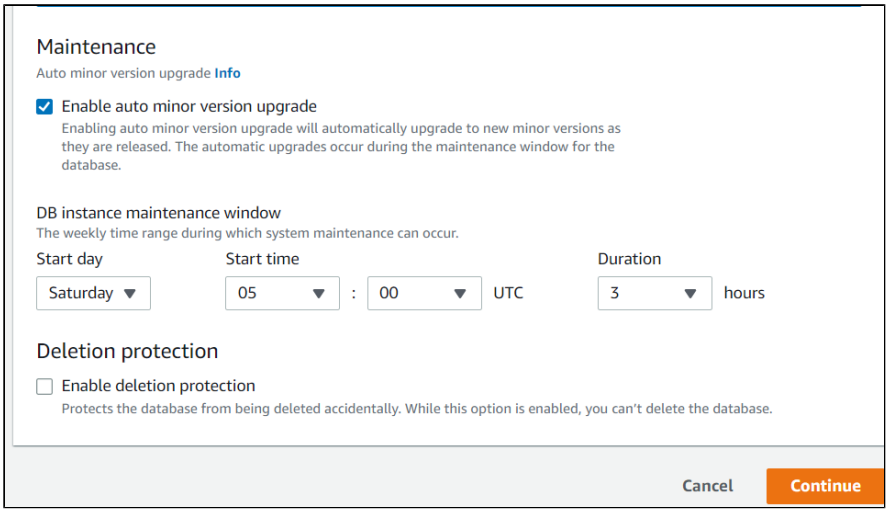
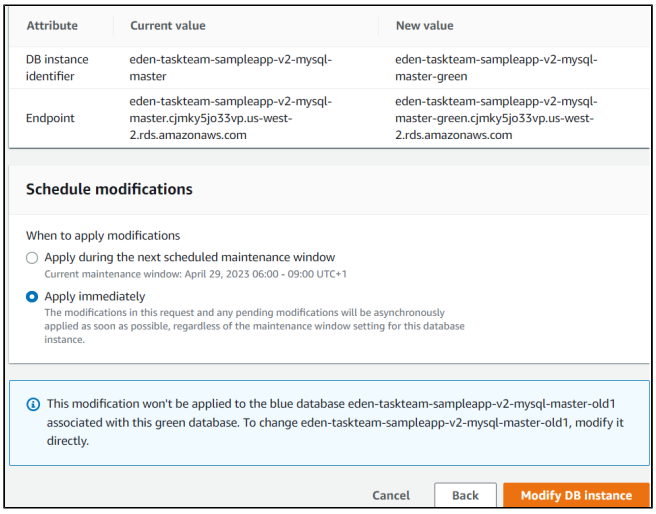
After switching over,

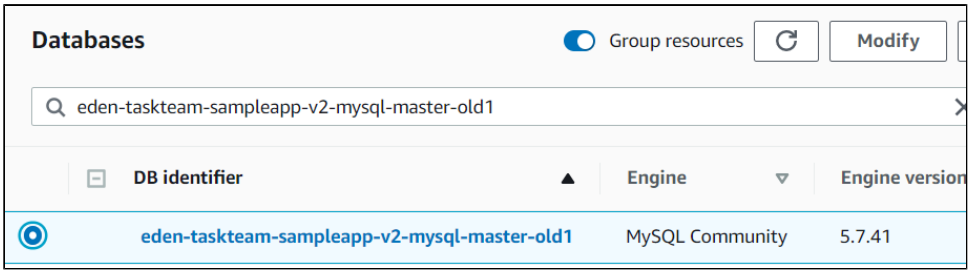
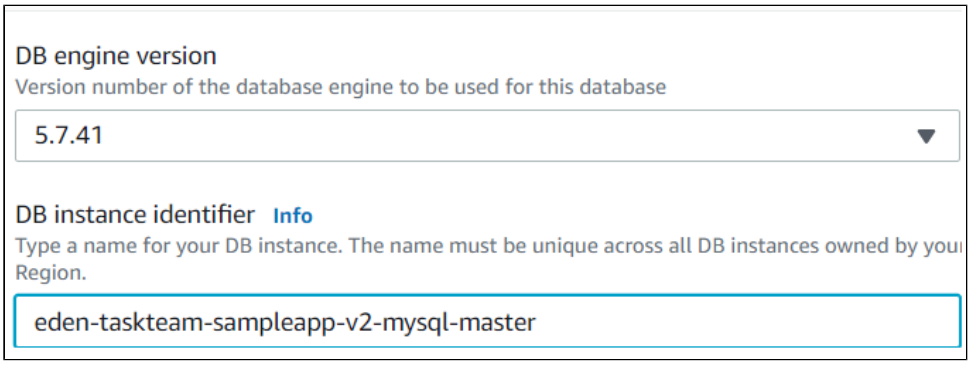
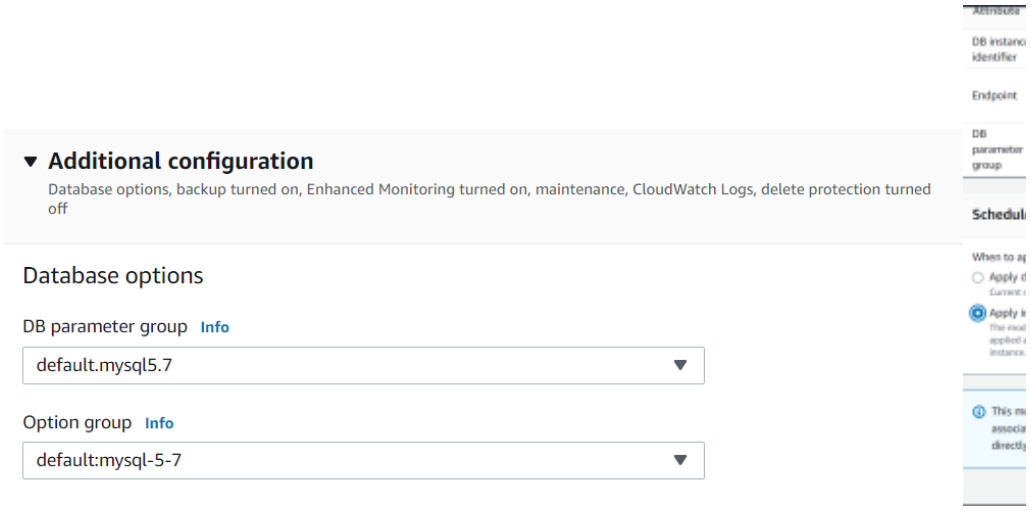
1.1 The blue instance name will be appended with **-old1.**

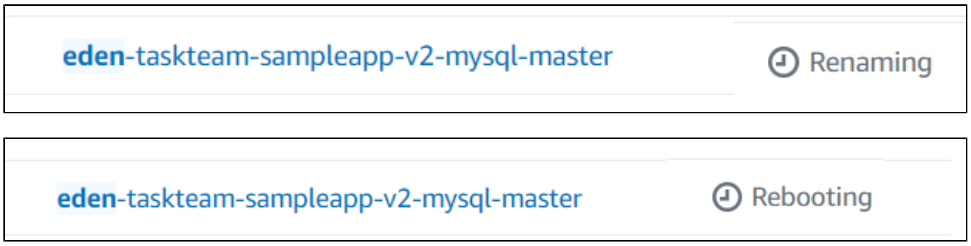
1.2 The green instance name will update with the name of the prod db instance. i,e it takes the name of the blue instance.



As part of the rollback, we need to reverse actions 1.1 and 1.2.

#	Description	Instructions	Screenshot
1	select the green instance	go to database dashboard and search the green instance db	
2	Modify the db instance indentifier	After selecting db instance, click on Modify and ONLY rename the green instance Note: Don't do any other action	
3	Go to next page	Click on Continue button at the end of the page	
4	Modify DB instance (Apply immediately)	After clicking on Continue, it shows the summary, and select Apply Immediately radio button. and Click Modify DB Instance button. This renaming takes 2 to 3 minutes	

5	Select the blue instance	go to database dashboard and search the blue instance db and continue the steps 3 and 4. It takes 2 to 3 mins to rename the db instance.	
6.	Modify the db instance identifier by removing the prefix **-old1**	After selecting db instance, click on Modify and rename the blue instance by removing the **-old1**	
7.	Change the DB Parameter group to mysql.default.5.7	Change the db parameter group mysql.default.5.7 Click on Modify.	

8. h	Modify DB instance (Apply immediately)	After clicking on Continue, it shows the summary, and select Apply Immediately radio button. and Click Modify DB Instance button. This renaming takes 2 to 3 minutes. Note: After renaming, it reboots the db instance. **Note: During renaming /rebooting, if the application connects to vault, it complains bad request, but pods are running successfully.**	

CAUTION: During the rollback, after the switch over whatever new data is written in green instances, will not be ported to old db instances. We are losing the data.

You can review approach shared in blog post from @Matkar, Rajesh which you could leverage

and customize https://aws.amazon.com/blogs/database/performing-major-version-upgrades-for-amazon-aurora-mysql-with-minimum-downtime/

Questions to AWS :

1. When the green instance will be available for writing ?(is any reboot required?)

The green instance will be available for writing once the switch over is successful.

7. Deleting a blue/green deployment

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments-deleting.html

8. FAQ