

- 🏠

Home
- 📖

Library
- 👤

Profile
- 📄

Stories
- 📊

Stats
- Following >
- 👤

Discover more writers and publications to follow.

[See suggestions](#)



使用 Blue-Green Deploy 把 Mysql 5.7 升級到 8.0

Justin Holly

Follow

 9 min read · Feb 11, 2024

👍 1

💬 1

🔖

📌

⋮

最近因為 Mysql 官方公佈 5.7 EOL 是 2023/10/01，AWS RDS 也公布他們只支援到 2024/02。我們公司算一算若不升級，會有一個保護費，每月要付 20 萬台幣左右。

其實很早就得知這個消息了，但是公司你懂的，不見棺材不掉淚，一兩年前說要升級，一定會跟你說，「升級有什麼明顯的效益嗎？」「升級可以賺錢嗎？」非得要等到真的要收錢了才來趕。

今天我就來分享如何在短時間盡可能覆蓋到所有整合測試的方式。如果你也是用 AWS RDS，那也許你可以參考一下。不過在我發文的這個時間，你如果還沒有升級完成，那你真的是有點危險。

為什麼要使用 Blue-Green Deployment?

初步評估後，我們使用 RDS 內建的藍綠部署升級功能。

原因很簡單，因為 AWS 幫我們處理了即時資料同步（blue-green sync）問題，再加上一鍵部署，並且幾乎 zero downtime。但是缺點是無法 roll back。部署完成後，會有一個備份 DB，只能手動轉移連線到那台備份 DB。

我們今年 SRE 離職，加上人事凍結，目前沒有專業的幫我們處理人工部署，所以使用託管服務的部署絕對是唯一最佳解。

AWS Blue-Green Deployment 簡介

由於**官網**已經有列出非常詳細的優缺點以及原理，我這邊就簡單的列出執行順序。

1. 按下 Blue-Green Deploy 按鈕後，會開始選擇一系列的參數，例如會不會希望升級至 mysql8.0 後要不要加大儲存空間，或是要不要修改 paramter group 等等，都完成後，就會啟動一個樹狀結構，如以下。藍色是我們目前的環境，而綠色是已經升級到 Mysql8.0 的環境，等待著切換

○	█	Blue	Available	Primary	MySQL Community	us-east-1d
○	█	slave2 Blue	Available	Replica	MySQL Community	us-east-1d
○	█	slave-1 Blue	Available	Replica	MySQL Community	us-east-1c
○	█	deploy	Available	Blue/Green Deployment	-	-
○	█	green-ksl2yo Green	Available	Primary	MySQL Community	us-east-1b
○	█	slave2-green-tf0ru8 Green	Available	Replica	MySQL Community	us-east-1d
○	█	slave-1-green-dsmvys Green	Available	Replica	MySQL Community	us-east-1c

2. 接下來由於我們的 service 都還是連線到藍色環境，RDS 都會藉由 binlog 同步到綠色環境。但需要注意的是假如綠色環境有異動，是不會被同步到藍色環境的。這邊需要非常注意，如果不小心更動藍色環境意料的外的資料，會導致資料衝突，甚至無法繼續下一步驟，因為 RDS 會隨時的檢查資料

官網特別提及這件事情，並且提醒綠色環境盡可能是 Read Only

3. 當使用者在綠色環境都做好測試之後，就可以按下 switch over 按鈕進行切換，RDS 會自動把原本的 endpoint 從藍色的 instance 轉到綠色的 instance。轉換過程很療癒，有一些可愛的小動畫。

最後轉換完成後，他會建議你刪除這個 Blue/Green Deployment task，並且留下一個同名同姓的 instance，但已經升級到 Mysql8.0了。

也別擔心舊的會不見，RDS 會以 `-old` 為 suffix 做一個複製給你備用，假如真的升級後發生問題，你可以在緊急把連線切回來。

○	█	Available	Instance	MySQL Community	us-east-1d	db.t3.medium
○	█old1	Available	Instance	MySQL Community	us-east-1d	db.t3.medium

以上就是官方介紹懶人包，就是如此簡單。

接下來要開始來升級了，根據我們團隊的服務使用 RDS 的情況，我們列出了比較擔心的幾個坑，以及我們的檢查步驟。

坑的評估

- 我們使用 NodeJS 開發後端，一堆 legacy 使用 Sequelize V3，因此很擔心會有語法與 Mysql 8.0 不相容，導致出現套件錯誤。這點需要整合測試檢查。

• character set(charset) / collation 設定不相容，由於 mysql8.0 預設變成 `utf8mb4`，需要檢查是否有資料是不相容的。

• storage engine type：下面會有表格列舉，基本上 mysql8.0 預設是 `innnoDB`

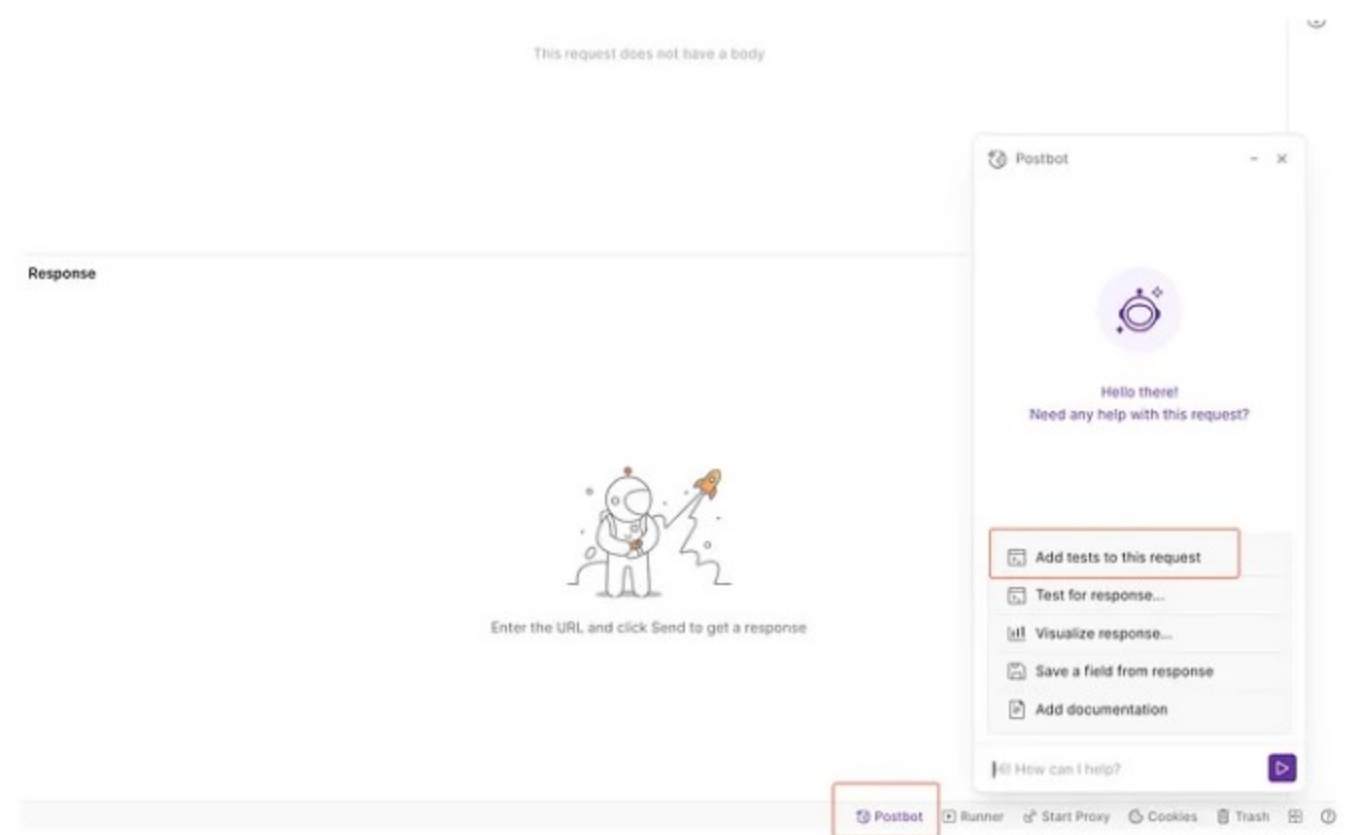
基於以上幾點我們內部比較擔憂的，整理出以下檢查事項。

升級步驟

- 1. 先使用官方的 SOP 檢查是否有不兼容的 configuration，可以在 DB 層級先調整。這部分其實如果不手動執行，RDS 也會自動幫你做，手動做單純是為了提早發現問題，可以先解決。
- 2. 先備份好 snapshot，點擊 RDS Blue-Green Deployment 部署，先挑一個使用量最大的 DB instance snap shot 做一個 dummy instance，走一遍 Blue-Green Deployment 確認流程沒有問題。
- 3. 準備 API 整合測試。這點是最為困難的，因為團隊過去沒有寫 API 整合測試，而在盤點過後我們總共有近百隻 API，好加在 postman 竟然在近期推出了 postbot AI 服務，可以根據 request/response 內容自動產生測試！！

因為時間真的很趕，我們的 API 大部分也都只是純粹的 CRUD。我認為這種產生 test case 的方式算是有一定的可靠度。當然金流相關的必須獨立測試，因為我們金流並沒有把測試環境獨立出來，會跟第三方耦合，而大部分的第三方（LinePay, TapPay）也都沒有特別設計 sandbox，即使像是 ApplePay, GooglePlay 等等有設計的，也跟我們前端綁死，後端無法獨立測試...

因此只好手動走一遍完整的購買，好在我們的 staging 環境是會自動退款的...



底下有個 postbot，可以幫你做很多事情



點完之後，就會自動產生 test cases

- 4. 開啟 Blue-Green Deployment，並且在綠色環境進行整合測試。這邊是我認為最關鍵的地方，因為如果在簡介所說，RDS 預設綠色環境是 Read-Only，那我要怎麼做 write 相關的整合測試？

這邊跟 AWS Support 討論過後定下一個策略就是：再開一台綠色環境的 instance replica，並且強制去改他的 parameter group，把 read-only 改成 0，也就是改成 writable 的意思。我們將這台 replica 命名為 green-for-inte-test。

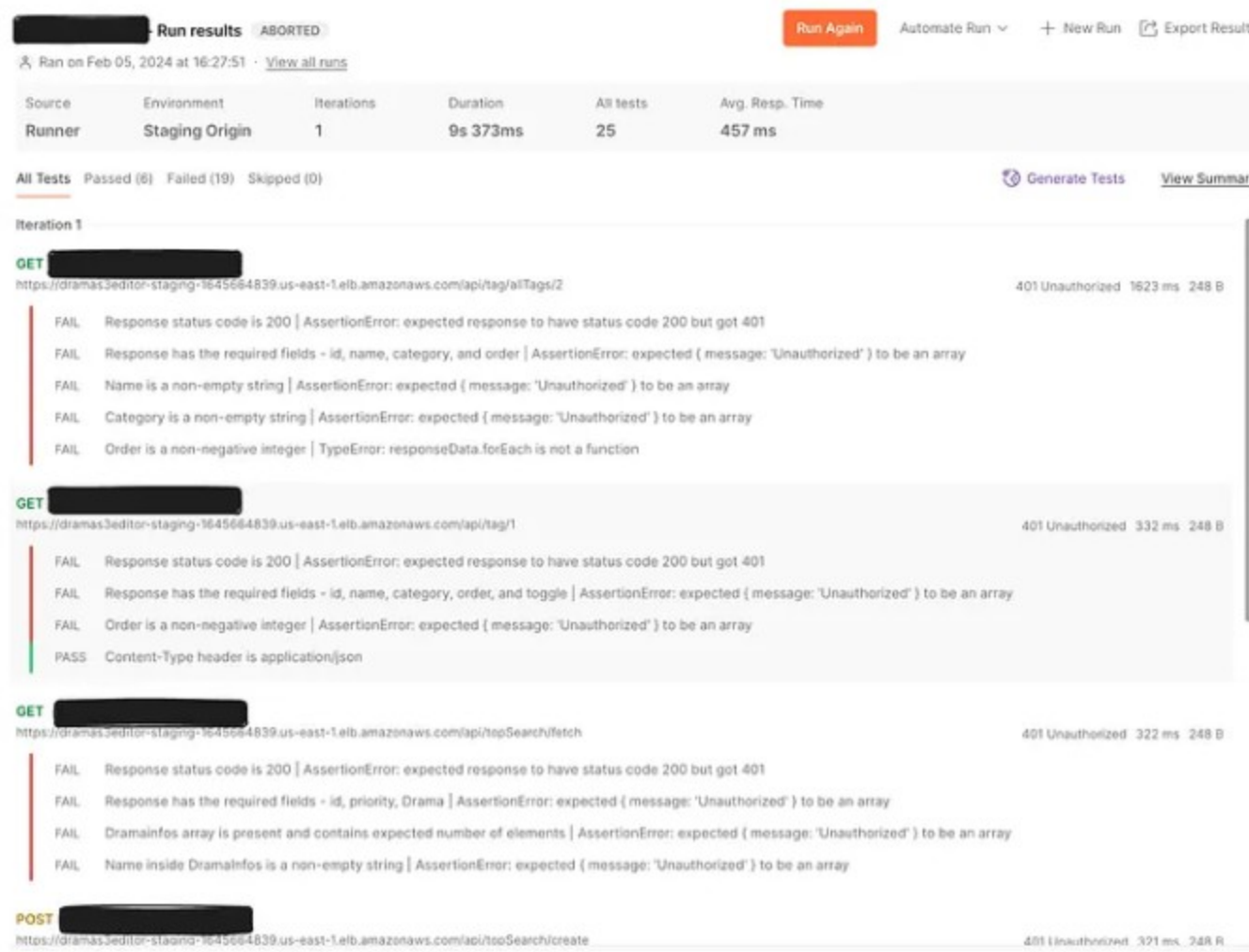
因為綠色環境的任何 instance 都不會影響到其他綠色環境，也不會影響到藍色環境，因此在這台 `green-for-inte-test` 做任何事情基本上都沒關係。我們只要在測試完成後，把他刪除即可。



原本只有兩台 slave-1, slave-2，多加一台 green-for-ine-test

這邊會做一些比較笨的事情，就是將我們的 services 連線都先暫時改成這台新的 DB，好險我們有寫 IaC 不然真的會改到哭暈。

改完之後就可以執行 postman 整合測試了，他們這個功能雖然簡單，但卻做得不錯。



一鍵就可以完成啦

其實以上提到的 char set, collation 等等的，會在一開始的 Mysql 檢查完後，這個階段繼續找。

例如 sequelize 套件語法跟 Mysql8.0 不相容，應該就會在 API 層級噴錯。

隱藏的問題我預期也是在這個測試階段可以找出來。假如真的找不出來，那我認為可能真的只能等到 production 噴錯才能發現了...畢竟能找的都找了..

5. 測試完畢後，記得刪除剛剛建立的 replica（`green-for-inte-test`），按下 switch over 就搞定。當然，剛剛改的連線資訊要記得改回來！

你以為這樣就結束了嗎？沒有！

我第一次執行 switch over 後就噴錯了，因為 data conflict！AWS 官方講的話真的要聽，不然就出事了，好在只是 staging 的 DB。當時的狀況是，我沒把連線資訊連到 `green-for-inte-test`，反而連到 slave-1，導致髒資料殘留，最後 switch over 的時候，RDS 比對發現資料不一致（primary key 重複，無法

藉由 binlog 寫入），就直接中斷 switch over。

好加在 RDS 還有一些即時判斷的功能，不然就真的把髒資料帶進去了。

這個問題，必須取消整個 Deployment，重來一遍才解決。

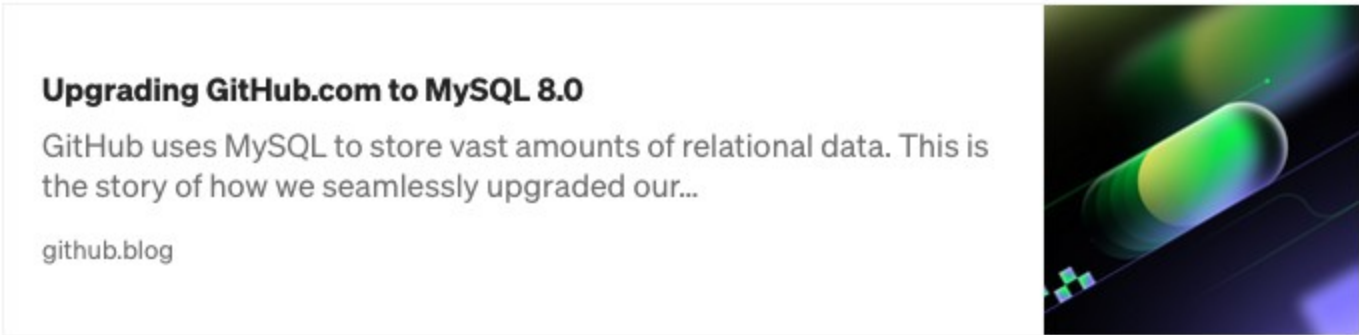
切換完畢後，如簡介所說，會留一下台以 -old 為 suffix 的機器，那台是我們的救命機器，如果真的有問題是在整合測試沒有發現的，又沒辦法即時 hot fix，就趕快把連線切到那台舊機器吧！

我們目前也只有部分 instance 升上去 8.0，預計是一季的觀察期，等到全部都升上去，然後都沒有問題就才會把舊機器移除。

反省

我認為說真的，這次 DB 升級是真的因為時間很趕，而沒有到很謹慎。

如果參考 Github 他們升級方式就會發現



他們還有做 read write 分批上線。

也許我們的數量沒有那麼大，也沒有做 shard，也沒有調整 DB 參數（幾乎都用 ORM，除了 CRUD 分析資料之外，很少直接下指令調整 DB 參數），所以相對沒有發生什麼問題。

但覺得他們分批上線的策略真的好酷啊！希望有機會也能參與一套完整的部署升級。

MySQLDeploymentBlue Green DeploymentDatabaseBackend

1

1

Written by Justin Holllly

15 followers · 27 following

寫下來才會記得

Follow

Responses (1)

Wnlin

What are your thoughts?

Captured by FireShot Pro: 12 8月 2025, 15:55:21

https://getfireshot.com



ERIC
Apr 22, 2024

...

這句話應該是指"綠色環境"嗎?

...

官網特別提及這件事情，並且提醒藍色環境盡可能是 Read Only

...



 1 reply

[Reply](#)

More from Justin Hollyly



 Justin Hollyly

Multi-Process(多行程)&Multi-Thread(多執行序)到底是個啥(1)
Multi-Process(多行程)&Multi-Thread(多執行序)到底是個啥(2)

Oct 10, 2021

 125



...



 Justin Hollyly

DAO, DTO 啥鬼？還有 GTO 嗎
由於是用 Javascript 開發後端，對於 design pattern 的概念本來就很薄弱，儘管使用...

Jun 20, 2022

 5



...



 Justin Hollyly

Multi Thread (多執行序) 到底是個啥 (3) 之 NodeJs 也有 threads
Multi-Process(多行程)&Multi-Thread(多執行序)到底是個啥(1)

May 31, 2024

 14



...



 Justin Hollyly

Multi-Process(多行程)&Multi-Thread(多執行序)到底是個啥(2)
Multi-Process(多行程)&Multi-Thread(多執行序)到底是個啥(1)

Jan 13, 2022

 2



...

See all from Justin Hollyly

Recommended from Medium



ThreadSafe Diaries

He Was a Senior Developer, Until We Read His Pull Request

When experience doesn't translate to expertise, and how one code review change...

Aug 3 5.1K 156



<devtips/>

You don't need 20 tools. Just use Postgres (seriously!)

One boring old SQL database might be the best backend in 2025.

Aug 2 669 5



Uzzal Kumar Hore

Beyond HA: A Battle-Tested PostgreSQL Architecture for...

In today's data-driven economy, PostgreSQL has become the backbone of mission-critical...

6d ago



Zudonu Osomudeya

AI, GitOps & DevSecOps: 2025 Skills That Hiring Managers...

DevOps Skills 2025: AI, GitOps & DevSecOps Jobs Guide for Beginners

4d ago 123



Young Gyu Kim

Traefik & Kubernetes—The Kubernetes Ingress Controller

Overview

Mar 25



Priyanshu Rajput

Why We Chose PostgreSQL Over MongoDB for Our High-Traffic Ap...

Intro: The Great Database Debate—SQL vs NoSQL

6d ago 3

See more recommendations