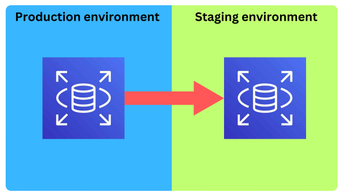
1


Introduction

1 RDS Blue/Green Deployment lets you create two environments: "blue" (current database) and "green" (updated database)
2 The blue environment is your live database currently serving your applications
3 The green environment is a duplicate where you can test updates like schema changes or engine upgrades
4 Once the green environment is fully tested, traffic automatically switches from blue to green
5 This ensures updates happen smoothly with minimal downtime and an option to roll back if needed

2 How Does Blue/Green Deployment Work?

1 A company is running their database on PostgreSQL 13 in the blue environment Represents your current production database instance or cluster
2 They want to upgrade to PostgreSQL 15 for new features and better performance
  1 They create a green environment with PostgreSQL 15, copy the data, and test it with their application
  2 Updates, such as schema changes or application upgrades, are applied to the "green" environment
3 Testing the Green Environment
  1 You can test the "green" environment by redirecting test traffic to it
  2 This ensures that updates function as expected before making it live
4 Switchover Once testing is successful, the Blue/Green Deployment feature automatically redirects production traffic to the "green" environment
5 Rollback Option If the "green" environment fails or has issues, the process allows switching back to the "blue" environment

3 Use Cases for Blue/Green

1 Database Engine Upgrades
2 Schema Changes
3 Testing New Features
5 Disaster Recovery and High Availability
6 Performance Testing

4 Limitation Of RDS Blue/Green

1 Temporary Cost Increase
2 Limited Database Engines Currently, Blue/Green Deployments are supported only for RDS for MariaDB, RDS for MySQL, and RDS for PostgreSQL.
3 Data Synchronization Overhead
4 Application Connection Handling
5 No Full Automation for Some Changes