

Blue/Green Deployment with RDS

In Blue/Green deployment with RDS firstly we must create two identical environments they are one is blue Environment (Production Environment) and other one is green environment (Staging Environment).

In this process several steps involved while we are creating RDS Blue/Green deployment we must follow these steps they are:

STEP:1

Blue Environment (Production Environment)

In Blue Environment (Production Environment) we have to create one dB instance that is mydb1 after creating mydb1 dB instance available state then we have to one create read replica for the mydb1 primary blue dB instance in blue environment (production environment) that name like mydb2. This mydb2 is the read replica for the mydb1 of the primary dB instance in the blue environment (Production Environment).

DB name: **mydb1 primary database instance (blue primary dB instance)**

Read Replica for mydb1 is: **mydb2**

Endpoint URL of the Blue Primary dB instance: **mydb1.cqcuvuhfhlhn.us-east-2.rds.amazonaws.com**

Then blue primary dB instance goes into available state after that select the primary dB instance is mydb1 and navigate through the actions and select the **create blue/green deployment-new** option now we must create another environment that is green environment (staging environment).

STEP:2**Green Environment (Staging Environment)**

By using Amazon RDS Blue/Green Deployments, you can make changes to the database in the staging environment without affecting the production environment. For example, you can upgrade the major or minor DB engine version, change database parameters, or make schema changes in the staging environment.

dB name: mydb1-green-7yrg2u (Green-Primary dB instance)

Read Replica for mydb1-green-7yrg2u is: **mydb2-green appended with some alphanumeric**

Endpoint URL of the Green Primary dB instance: **mydb1-green-7yrg2u.cqcuvuhfhlhn.us-east-2.rds.amazonaws.com**

STEP:3**EC2 instance**

Meanwhile we have to create one ec2 instance by using ec2 instance. After entering some commands those commands are as shown below. We can connect the databases in the blue environment and green environment and also, we can see practical experience in ec2 machine by using both environments endpoint URL we can access both environments' databases.

In EC2 instance we are using the existing security group name as **rds demo**.

Security group for both RDS dB instances and EC2 instance we are using same security group that is name as rds demo.

The security group inbound and outbound rules are as shown in below:

Inbound rules are:

Type	Protocol	Port Range	Source
HTTP	TCP	80	Custom 0.0.0.0/0
HTTPS	TCP	443	Custom 0.0.0.0/0
SSH	TCP	22	Custom 0.0.0.0/0
MYSQL/Aurora	TCP	3306	Custom 0.0.0.0/0
All traffic	All	All	Custom 0.0.0.0/0

Outbound rules are:

Type	Protocol	Port Range	Source
All traffic	All	All	Custom 0.0.0.0/0

Commands are used in after launching EC2 machine:

```
sudo su -
```

```
yum update -y
```

```
yum install mysql -y
```

```
Mysql -h endpoint URL of the Blue Environment (Production Environment) -u admin -P 3306 -p
```

then click on enter button now we enter the password then only we are connected to the databases in blue environment.

After that we can enter some commands in mysql those commands as shown in below

```
show databases;
```

```
create database blueenvironmentproduction;
```

Now one database is created in blue environment that is **Blue Environment (Production Environment)**

Now we enter command **show databases;**

After that enter one command is **exit;**

After entering the exit command, we can exit from the present environment i.e., blue environment (production Environment)

Now we must connect to the green environment (Staging Environment) by using endpoint URL of the green environment (Staging Environment) with the help of EC2 instance.

Mysql -h endpoint URL of the green Environment (staging Environment) -u admin -P 3306 -p

then click on enter button now we enter the password then only we are connected to the databases in green environment.

After that we can enter some commands in mysql those commands as shown in below

show databases;

Previously created one database in blue environment that is also reflected into the green environment

create database **greenenvironmentstaging;**

Now one error is occurred in green environment that is as shown in below

ERROR 1290 (HY000): The MySQL server is running with the --read-only option so it cannot execute this statement

Now enter command **show databases;**

After that enter one command is **exit;**

After entering the **exit** command, we can exit from the present environment i.e., green environment (staging Environment)

Note: Because we are unable to write permissions in a green environment, we can only read permissions. However, we can carry out both Read and Write Operations in a Blue Environment.

Step 4:

switch over

Once the testing has been completed on the green environment, live application traffic is directed to the green environment and the blue environment is deprecated.

Before you switch over production traffic is routed to the databases in the blue environment.

After you switch over, production traffic is routed to the databases in the green environment.

Switch over time period between 30 sec to 3,600 Sec. The default timeout period is 300 sec.(5min)

Now go to RDS dB instances then Navigate through rdsbluegreendeployment after that go to actions and click on switch over.

NOTE: After a switchover, the DB instances in the previous blue environment are retained. Standard costs apply to these resources. DB renames the DB instances in the blue environment by appending -old *n* to the current resource name, where *n* is a number.

Now mydb1 and mydb2 are the production environment database instances in the green environment and mydb1-old1 and mydb2-old1 are the staging environment database instances in the blue environment that means traffic is shifted to blue environment to green environment.

Now we can see the endpoint URL different in both environments after completion of switch over the endpoints are like this as shown in below.

Before Switchover:

Blue Environment (Production Environment)

Endpoint URL: mydb1.cqcuvuhfhlhn.us-east-2.rds.amazonaws.com

Port:3306

DB instance ID: mydb1

Engine version:5.7.39

Green Environment (Staging Environment)

Endpoint URL: mydb1-green-7yrg2u.cqcuvuhfhlhn.us-east-2.rds.amazonaws.com

Port:3306

DB instance ID: mydb1-green-7yrg2u

Engine version:8.0.32

After switchover:

Green Environment (production Environment)

Endpoint URL: mydb1.cqcuvuhfhlhn.us-east-2.rds.amazonaws.com

Port:3306

DB instance ID: mydb1

Engine version:8.0.32

Blue Environment (Staging Environment)

Endpoint URL: mydb1-old1.cqcuvuhfhlhn.us-east-2.rds.amazonaws.com

Port:3306

DB instance ID: mydb1-old1

Engine version:5.7.39

Note: After completion of switchover, we are unable to write permissions in a blue environment (staging environment), we can only read permissions. However, we can carry out both Read and Write Operations in a Green Environment (Production Environment).

Advantages of Blue-Green Deployment with Amazon RDS:

- 1.Minimal Downtime:** Blue-green deployment allows you to deploy a new version of your application alongside the existing version, without interrupting the availability of your application. This ensures minimal downtime during the deployment process.
- 2.Rollback Capability:** With blue-green deployment, if any issues or bugs are discovered after the new version is deployed, you can easily roll back to the previous version by redirecting traffic back to the blue environment. This helps to mitigate risks and maintain application availability.
- 3.Simplified Testing:** Blue-green deployment allows for extensive testing of the new version in a production-like environment. You can thoroughly test your application against the green environment before routing traffic to it, ensuring better quality control and reducing the chances of introducing errors into the production environment.
- 4.Seamless Switching:** Once the new version has been successfully deployed and tested, you can seamlessly switch traffic from the blue environment to the green environment. This enables you to make the new version instantly available to users without any disruption.

Disadvantages of Blue-Green Deployment with Amazon RDS:

- 1.Increased Infrastructure Complexity:** Implementing blue-green deployment requires maintaining duplicate infrastructure environments (blue and green) to run both the current and new versions simultaneously. This adds complexity and may increase infrastructure costs.
- 2.Data Synchronization:** When using Amazon RDS, ensuring data consistency between the blue and green environments can be challenging. Data changes made during the deployment process need to be replicated to both environments to maintain data integrity.
- 3.Resource Utilization:** Running two parallel environments means that you need to allocate additional resources to support both versions simultaneously. This can increase resource utilization and may result in additional costs, particularly if the green environment needs to be provisioned with the same level of resources as the blue environment.
- 4.Operational Overhead:** Managing two environments simultaneously requires additional operational overhead. It involves monitoring, maintaining, and syncing the blue and green environments, which can add complexity to your deployment process.

Conclusion:

Overall, blue-green deployment with Amazon RDS offers several benefits such as minimal downtime, rollback capability, and simplified testing. However, it also introduces complexities related to infrastructure management, data synchronization, resource utilization, and operational overhead. It is essential to carefully consider these factors and assess whether blue-green deployment aligns with your specific application requirements and operational capabilities.

Please check the below link for further information:

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

Emp Name: Sai Kumar Vemula

**Blue/Green Deployment With
RDS**

Emp Id:6778