Migration Process of On-Prem Databases to Amazon RDS

If you are reading this blog, it means you are totally bored of what is out there on the web and you just want to get over with migrating your databases onto Amazon RDS.

I’m assuming you have conceptual knowledge of subnet groups, security groups, working knowledge of Amazon S3 like creating buckets, uploading objects, etc. and finally managing RDS instances. Those who have no experience with RDS and S3 will have to update themselves before reading this further. This blog is specifically about migrating databases to RDS. In our case I’m going to take example of MS SQL Server databases to migrate to RDS.

After you have successfully launched your RDS Instance, there are 2 main ways to get your database into RDS. The first one is via scripting and the second is using SQL backups. Which route to go is totally your personal choice. My personal choice is SQL backups. So, without further ado let’s see how we can migrate our databases to Amazon RDS service.

**Method #1: Using Scripts to migrate SQL data to RDS**

*Step 1:* This step is optional. For safety purpose take a snapshot of your source RDS instance

*Step 2:* This is optional too. For faster performance of migration process, you will have to disable automatic backups on the origin RDS instance.

*Step 3:* Create your target database and disable all foreign key constraints and triggers.

*Step 4:* Create Logins: Go to SSMS (SQL Server Management Studio), select your Source database, right click and select - All Tasks/Generate Scripts to create all logins. Then in SSMS, connect to your RDS Instance by using its endpoint, select your Target database and execute the generated script.

*Step 5:* Create Schema: Go to SSMS, select your Source database, right click and select - All Tasks/Generate Scripts to create all schema DDL. Then, select your Target database in SSMS and execute the generated script.

|  |
| --- |
| *Step 6:* At this stage you have to ways to transfer your data from source database to target database. You can either use the bcp (bulk copy command), or the Import/Export Wizard in SSMS.*Step 7:* Once data is transferred, re-enable the foreign keys and triggers.  |
| Step 8: If you had disabled the automatic backups on RDS Instance as in Step 2, re-enable them. |
|  |

Voila! You are done. But if you do not like this approach you can try Method #2.

**Method #2:**

**Using database backups files (.bak files) to migrate SQL data to RDS**

*Step 1*: Very first thing, go to your S3 service and create a new bucket (example: bakfiles) and upload your object (example: SampleDB.bak file).

*Step 2*: Go to your RDS Service and from left menu list select “option groups”. “Create Option Group” and give it a name, for example “sample-optiongroup” and assign appropriate engine: (sqlserver-se in our case).

*Step 3*: Select/check the given name of option group (sample-optiongroup), you will notice that SQLSERVER\_BACKUP\_RESTORE option is automatically defaulted and it is read only. Assign new IAM or Specify existing IAM role to allow your RDS to access your S3 Bucket and also point to your target bucket where .bak files are uploaded. This IAM account should be able to reach to your S3 buckets.

*Step 4:*Go back to your RDS Service, select your RDS Instance, select Modify from Instance Actions, scroll down to Database Options and change the Option Groups to the option groups you just created (in our case sample-optiongroup)

*Step 5:*Go to SSMSconnect to your RDS Instance by using its endpoint, open new query tab and exec below procedure. The first parameter is database name you have or want to call it in your target database in your RDS Instance. The second parameter is the ARN (Amazon Resource Name). You can find your ARN name in the bucket policy of your bucket. Go to your S3 bucket/Permission/Bucket Policy.

|  |
| --- |
| exec msdb.dbo.rds\_restore\_database |
|  @restore\_db\_name='NAME-OF-YOUR-DB', |
|  @s3\_arn\_to\_restore\_from='arn:aws:s3:::NAME-OF-YOUR-BUCKET/NAME-OF-YOUR-BAK-FILE.bak' |
| Example:exec msdb.dbo.rds\_restore\_database |
|  @restore\_db\_name='SampleDB', |
|  @s3\_arn\_to\_restore\_from='arn:aws:s3:::ifi-datasets/IRS/bakfiles/SampleDB.bak' |

The above is special procedure provided by Amazon RDS service to us, which will help us restoring the .bak files we uploaded on S3. There are other special procedures which will also help us as shown below.

|  |
| --- |
| rds\_backup\_database – This backs up a database to a S3 bucket. |
| rds\_restore\_database – This restores a database from S3 bucket. |
| rds\_task\_status – This will track all running backup and restore tasks. |
| rds\_cancel\_task – This can be used to cancel a running backup or restore task. |

For more information on migration of on-premise SQL Server databases to Amazon RDS, please contact me on my email address Tabish.Sayed@Informa.com.