



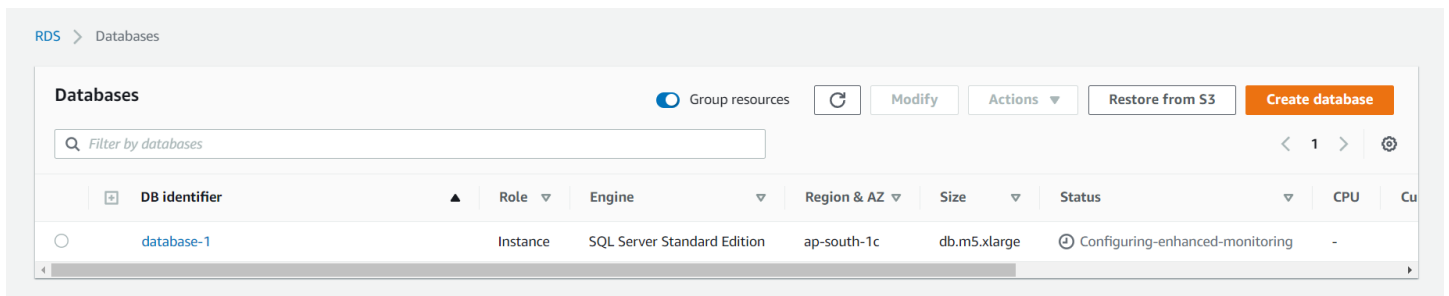
8/5/2022

Project - 04

Migrate a database server from on-premises to AWS.

Sumit Mishra
SIC: 190310286

1. Create a RDS server on AWS using the console.



2. Create an EC2 instance (Amazon Linux with ms SQL server) using the console.

Instances (1/4) Info									
Search									
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D	
<input type="checkbox"/>	vpc-1-server	i-077cf725cd8e4e932	Terminated	t2.micro	-	No alarms	ap-south-1a	-	
<input checked="" type="checkbox"/>	db-server	i-09eac4a3e59b936d0	Running	t3.xlarge	2/2 checks passed	No alarms	ap-south-1c	ec2-3-6-7-229	
<input type="checkbox"/>	vpc-2-server	i-0f245717dd3d42948	Terminated	t2.micro	-	No alarms	ap-south-1b	-	
<input type="checkbox"/>	vpc-3-server	i-0c9a772772e6d52e	Terminated	t2.micro	-	No alarms	ap-south-1b	-	

3. Configure SQL Server on the EC2 instance.

- While the server was launching, I created and attached an extra EBS volume to store the Database data.
- Connected to the EC2 instance using xShell and typed the following commands for configuring the SQL server.
 - Switched user to super user.
 - Stopped the SQL server.
 - Ran an mssql-conf script to reset the SA password.
 - Formatted the volume to ext4 type.
 - Created a directory name 'SQLServerData'.
 - Mounted the directory into the attached volume.
 - Attached some permissions to the directory.
 - Changed some settings using some commands to enable the SQLServerData director as the default data directory.
 - Restarted the ms-sql service.
- Connected the ms-sql server using the Microsoft SQL Server Management Studio.
 - Provide username as sa.
 - Provide password for the system server.
 - Select SQL server authentication.
 - Click connect.
- Created a database and some schema inside using the Microsoft SQL Server Management Studio.
 - Created a database using 'Create Database' Command.
 - Created a sample table inside the database.

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```


c. Inserted some records into it.

```
INSERT INTO Persons(PersonID, LastName, FirstName, Address, City)  
VALUES  
(1, 'Mishra', 'Sumit', 'Barmunda', 'BBSR'),  
(2, 'Mishra', 'Amit', 'Barmunda', 'BBSR'),  
(3, 'Mishra', 'Subrat', 'Barmunda', 'BBSR'),  
(4, 'Mishra', 'Mamata', 'Barmunda', 'BBSR'),  
(5, 'Mishra', 'Nabneet', 'Barmunda', 'BBSR');
```


5. Migrating the on-premises database to AWS using DMS service.

1. Created a replication instance for migration of Databases using port number 1433 for mssql.

DMS > Replication instances

 **Upgrades to versions 3.4.7 and higher**
You have 1 instance that uses AWS DMS version 3.4.7. Upgrades to AWS DMS versions 3.4.7 and higher require that you configure AWS DMS to use VPC endpoints or use public routes. This requirement applies to source and target endpoints for these data stores: S3, Kinesis, Secrets Manager, DynamoDB, Amazon Redshift, and OpenSearch Service. [Learn more](#)

[View endpoints](#)

Replication instances (1)  [Actions](#) [Create replication instance](#)

<input type="checkbox"/>	Name	Status	VPC	Class	Engine version	Availability zone	Public	Public
<input type="checkbox"/>	sql-replication-instance	Creating	vpc-01d987e17c3785149	dms.t3.medium	3.4.7	ap-south-1c	Yes	

2. Created a source server endpoint using the source server IP (Linux server's public IP) and tested it.

DMS > Endpoints > linux-sql-server > Test endpoint connection

Test endpoint connection

Replication instance
A replication instance performs the database migration

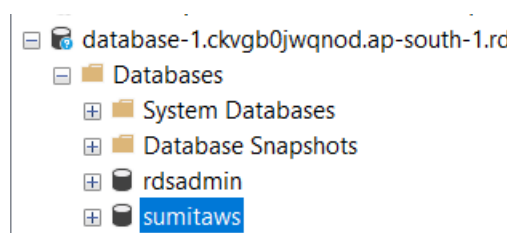
sql-replication-instance

[Run test](#)

Endpoint identifier	Replication instance	Status	Message
linux-sql-server	sql-replication-instance	successful	

[Back](#)

3. Connected to the RDS from Microsoft SQL Server Management Studio and created a database inside it.



4. Created the destination server endpoint using the RDS and tested it.

Run test

Endpoint identifier	Replication instance	Status	Message
database-1	sql-replication-instance	successful	

Cancel

Create endpoint

5. Create a database migration task.
 - a. Selected the created replication instance.
 - b. Selected the created source server endpoint.
 - c. Selected the created destination endpoint.
 - d. Assigned % in selection rules to enable full data migration.
 - e. Added tags and created the task.

DMS > Database migration tasks

Database migration tasks (1)										Actions	Quick view and compare	Create task
Find database migration tasks												
<input type="checkbox"/>	Identifier	Status	Progress	Type	Source	Target	Replication instance	Started	Stopped			
<input type="checkbox"/>	sql-replication-task	Creating		Full load	linux-sql-server	database-1	sql-replication-instance	-	-			

6. Waited for the migration task to fully complete, then tested using the Microsoft SQL Server management Studio to check if the contents of the on-premises server have been fully transferred into the RDS server on AWS.

DMS > Database migration tasks

Database migration tasks (1)										Actions	Quick view and compare	Create task
Find database migration tasks												
<input type="checkbox"/>	Identifier	Status	Progress	Type	Source	Target	Replication instance	Started				
<input type="checkbox"/>	sql-replication-task	Load complete	100%	Full load	linux-sql-server	database-1	sql-replication-instance	August 7, 2022 at 14:00				

```
select * from Persons;
```

380 %

 Results  Messages

	PersonID	LastName	FirstName	Address	City
1	1	Mishra	Sumit	Barmunda	BBSR
2	2	Mishra	Amit	Barmunda	BBSR
3	3	Mishra	Subrat	Barmunda	BBSR
4	4	Mishra	Mamata	Barmunda	BBSR
5	5	Mishra	Nabneet	Barmunda	BBSR

Using the above mentioned steps, I successfully completed the Database Migration Task from on-premises server to AWS.