Database Migration with AWS

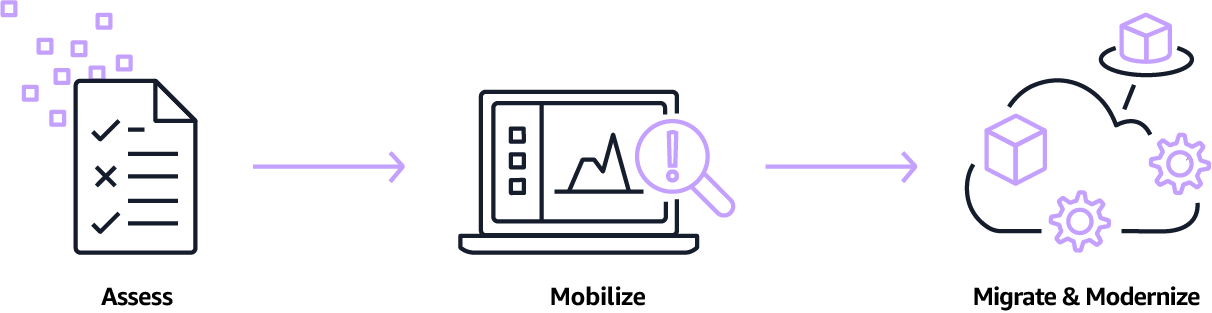
[Replatform – AWS DMS]

|  |
| --- |
|  |

**Introduction**

Migrating to the cloud

**Standard Migration Process**



**Project Description – Scenario**

In this project I will migrate a fictional application to AWS cloud. I will deploy the source environment with a CloudFormation template in my AWS account. The deployed resources consist of two t3.micro EC2 machines (one for webserver, one for database), a NAT Gateway, an API Gateway and two AWS Lambda functions (for easy retrieval of EC2 Access Key).

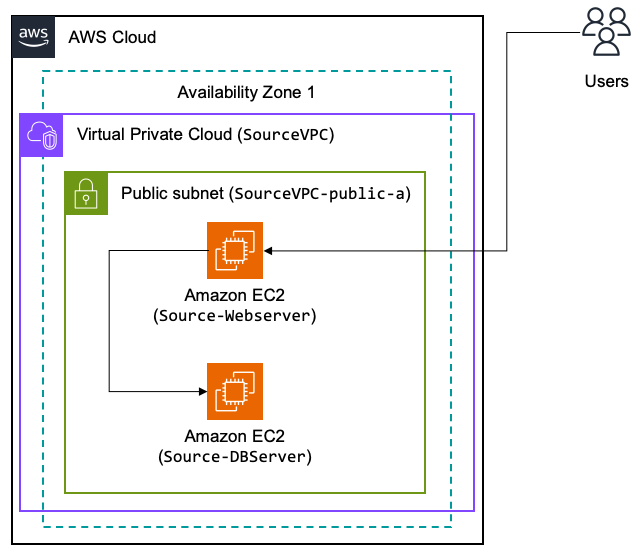
**Source Environment Deployed**

A screenshot of a computer

Description automatically generated

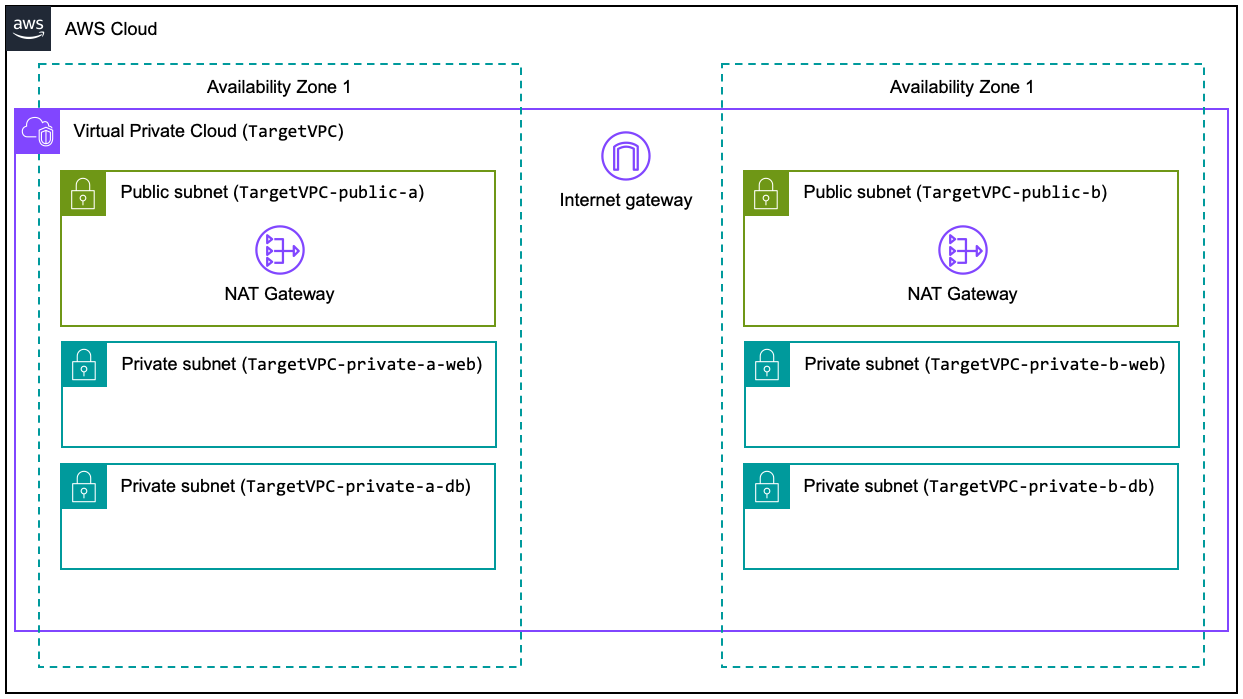
**Source Environment** **Architecture – Overview**

The source environment will be simulated by a segregated Amazon VPC named SourceVPC



**Target Environment** **Architecture – Overview**

The migrations are going to move the workload components from the source to the target environments – current status mostly empty.



**Discovery [AWS Migration Hub]**

For this project I will perform discovery of the on-premises environments by installing AWS ADS agent on each server and then review and analyse the discovered data.

**Verify ADS Credentials**

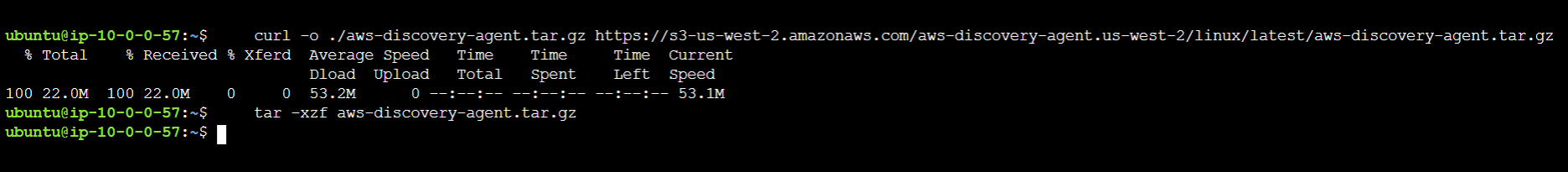
In order to deploy the ADS Agent into our Source-Webserver and Source-DBServer, we need AWS Identity and Access Management (IAM) credential with proper permissions to communicate with AWS ADS APIs.

A screenshot of a computer

Description automatically generated

**Deploy**

Install discovery agent on webserver **instance [Repeat Installation steps for DB Server and replace the AWS\_key\_ID and AWS\_secret\_key]**



install the agent and pass the appropriate IAM user credentials to the agent. Replace the AWS\_key\_ID and AWS\_secret\_key variables.

A computer screen with text

Description automatically generated

Verify that the discovery agent is started and analyzing the instance. **active (running).**

A black screen with text

Description automatically generated

Agent is now registered with the Application Discovery Service

A screenshot of a computer

Description automatically generated

A diagram depicting how the Webserver and the DBServer are all interrelated, and their dependencies. **[AWeSome!!!]**

A computer screen shot of a network

Description automatically generated

Agent-based discovery can be performed by deploying the AWS Application Discovery Agent on each of your VMs and physical servers It collects static configuration data, detailed time-series system-performance information, inbound and outbound network connections, and processes that are running.

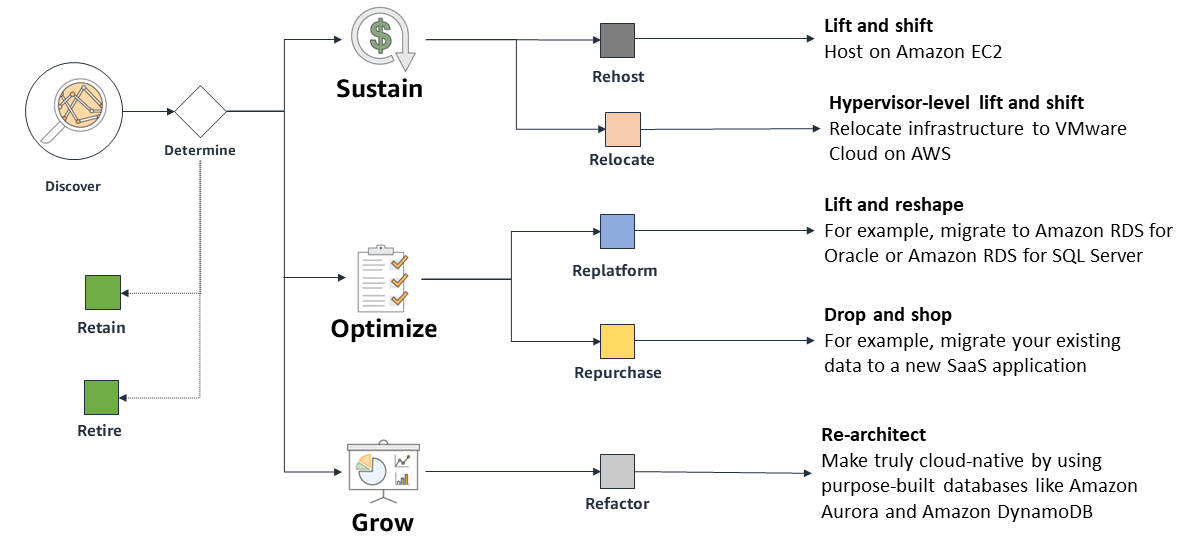
**Applications**

Some discovered servers might need to be migrated together to remain functional, you can logically define and group discovered servers into applications.

A screenshot of a computer

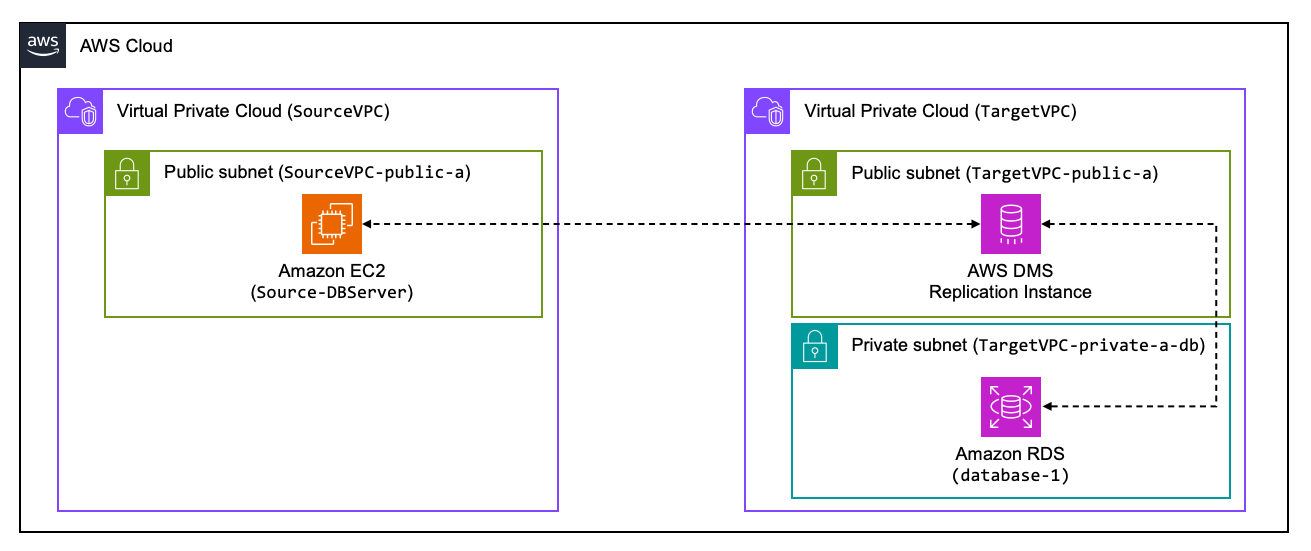
Description automatically generated

**Some Database Migration Strategies**



**Network Set-up**

The AWS Database Migration Service (DMS) Replication Instance, will be provisioned in a public subnet of the TargetVPC, will need to connect to the source database over public internet. The communication between the DMS Replication Instance and the target database will be done in private, within the TargetVPC.



**Security Groups**

 Create the Security Groups for the DMS Replication Instance

A screenshot of a computer

Description automatically generated

**Subnet group for the Target Database**

**A screenshot of a computer

Description automatically generated**

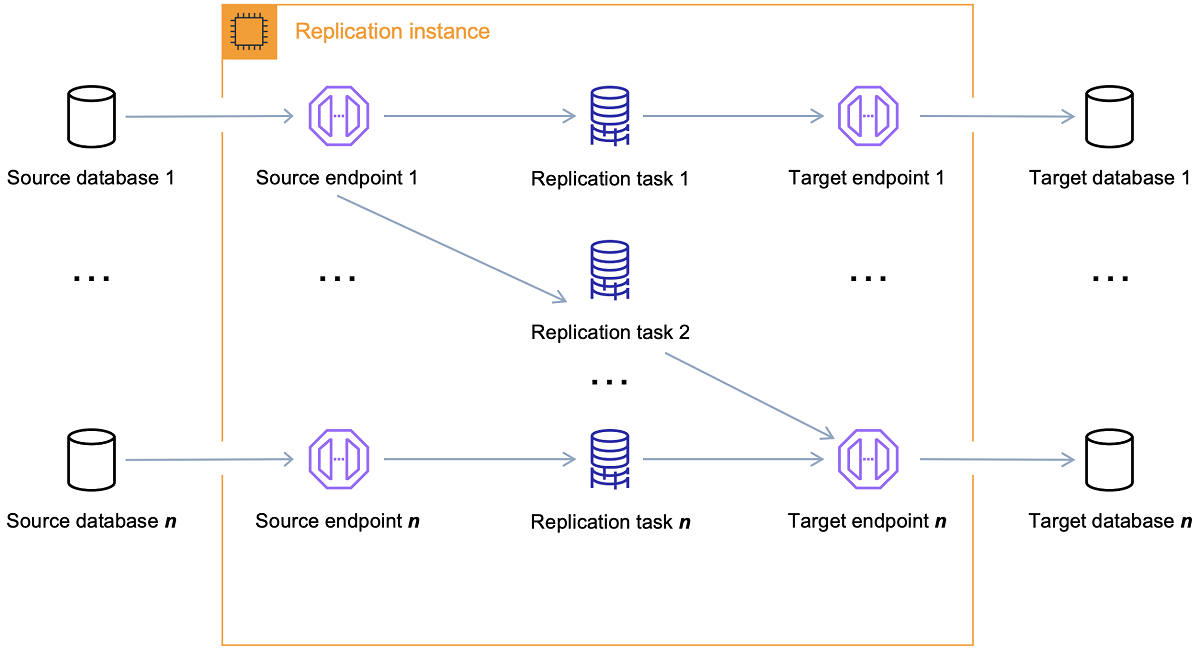
**Target Database**

A screenshot of a computer

Description automatically generated

**Replication Instance**

This instance will be used toprepare the resources needed for the migration with AWS Database Migration Server (DMS).



**Replication Subnet group [AWS DMS]**

The DMS Replication Instance requires a Subnet Group. This will control which subnets will be used to provision the managed EC2 instance.

A screenshot of a computer

Description automatically generated

**Replication Instance**

Will be used to migrate the data from the source database to the managed target database on Amazon RDS.

A screenshot of a computer

Description automatically generated

Updating the inbound rule of the DBServer Security Gorup and adding the **type** MYSQL/Aurora and **source** Public IP address assigned to DMS Replication Instance is required because the source database runs on a different VPC.

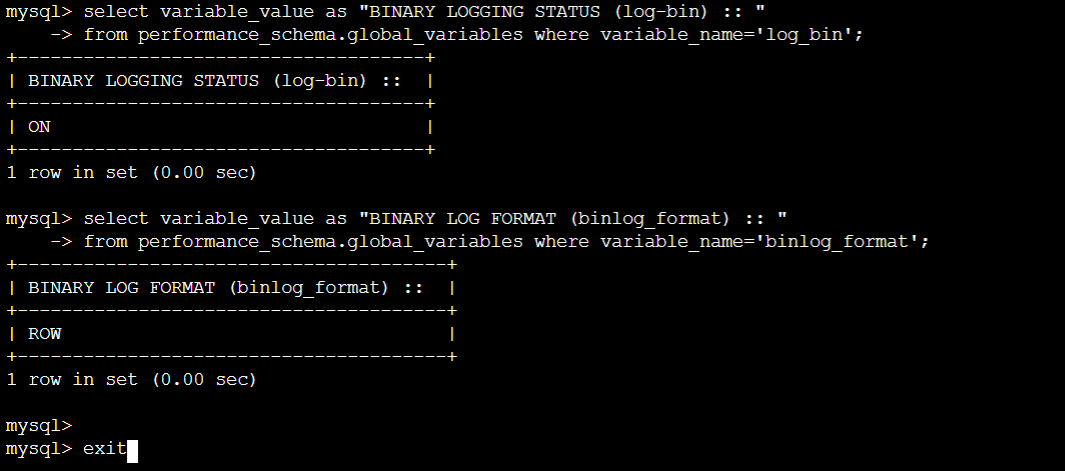
**Configure Source Database**

To ensure minimal downtime for the database migration, we're going to use continuous replication of changes, also known as Change Data Capture (CDC), from the source database to the target database.

**Enable CDC on source database**

For AWS DMS continuous replication with MySQL database as a source, you'll need to adjust the configuration of the source database. For that, we will connect to the server through a virtual terminal

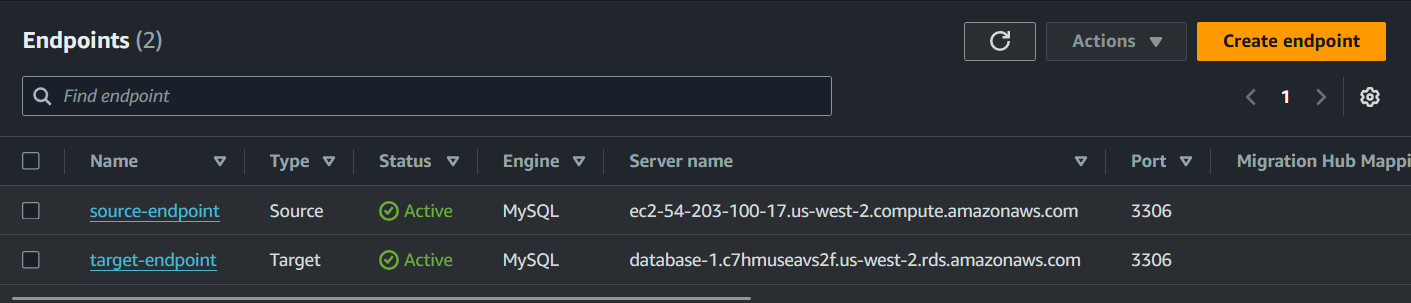
**Changes took effect after running a few commands**

****

**Connections from DMS to source and target databases! – Source & Target Endpoints**

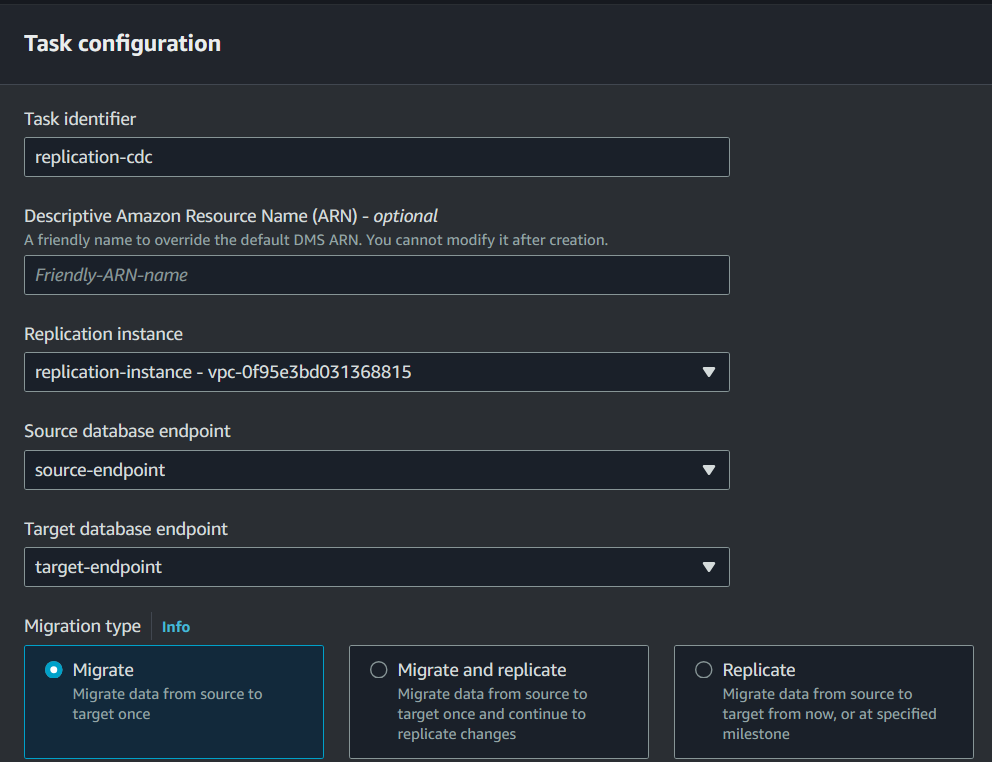
AWS DMS uses an endpoint to access your source or target data store, to then execute the replication tasks

**Endpoints**

****

**Configure & Run Replication Task**

You use an AWS DMS replication task to move a set of data from the source endpoint to the target endpoint.

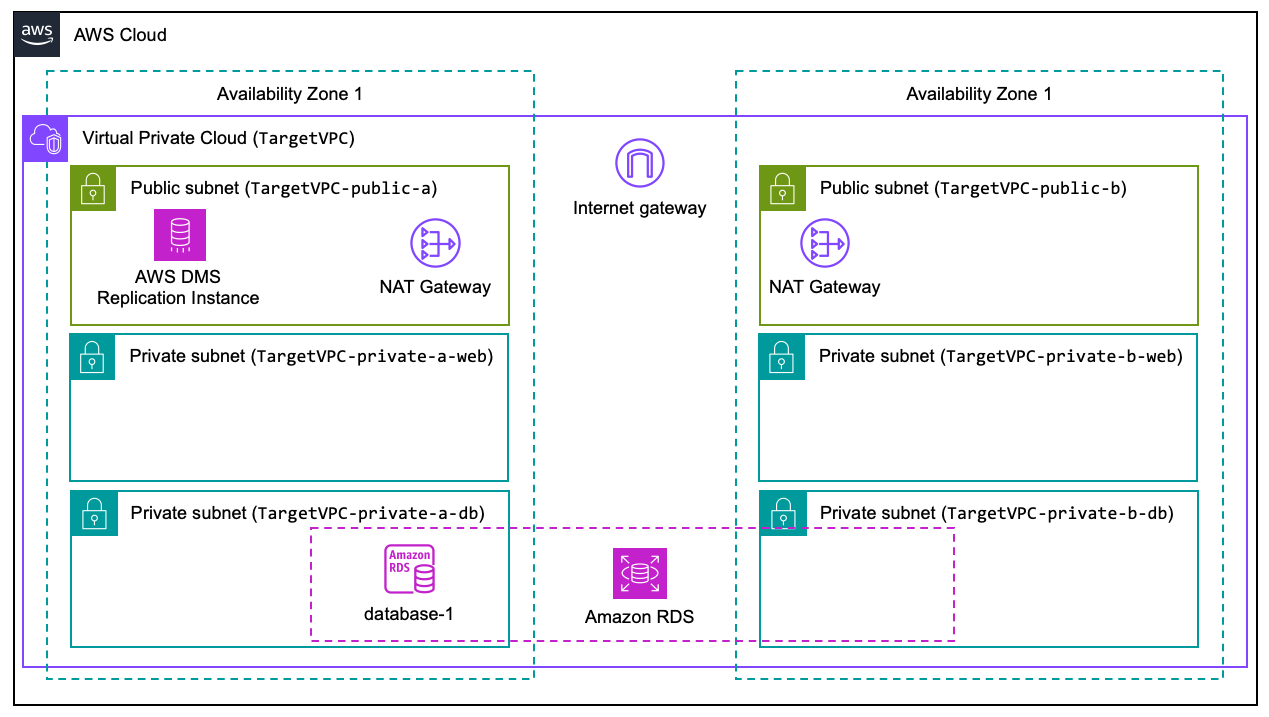
****

migrated the data from the source database server into our target managed database instance!

A screenshot of a computer

Description automatically generated

**Current Target Environment**



**Conclusion | Re-cap**

1. Created a new managed database, using Amazon Relational Database Service (RDS)
2. Created an AWS Database Migration Service (DMS) Replication Instance - that allows you to replicate data between databases
3. Created the source and target DMS Endpoints
4. Modified the configuration of the source database to allow for continuous replication of data (binary log)
5. Started the replication of data through a DMS replication task

**Done!**