AGILISIUM

BIG ON CLOUD. BIG ON DATA.

Guide:

Best Practices for Database Migration to AWS

aws

PARTNER
Advanced Tier
Services

www.agilisium.com

Migrating a database to AWS takes time and careful planning. It isn't just the database that needs migration; everything that uses it will need some changes. Sometimes it's just a matter of reconfiguration; in other cases, significant coding changes are necessary. The migration of the database itself may be a simple port with few changes, or it may require a new design.

What's necessary depends on which of Amazon's many options you use and how much will change from your existing database. This article looks at the best practices for migrating to an AWS database. It assumes you've already chosen the best destination and need to figure out the best way to do it. If you follow the best practices before, during, and after migration, you'll get it done relatively quickly and with few problems.

You can install almost any database you want under EC2, but then you must manage all its details. Here we assume you're moving to a managed database, which will be Aurora or RDS. Amazon management makes maintenance easier, but it will require doing some things differently.

Preparation for migration

You need to decide at the start what type of migration to perform.

- **Lift and shift:**
  This is the simplest approach. The AWS database migration consists of moving the database to the cloud with an equivalent engine and schema. The applications that use it remain the same except for configuration parameters.

- **Replatforming:**
  The new database includes changes to improve performance in the cloud, without changing the structure. Changes in indexing are often a part of this. Replatforming takes more effort than lift and shift, but it will produce better results.

- **Refactoring:**
  Changes are made to the database schema or stored procedures. They can be minor or sweeping. Applications will need to change to be compatible. This provides the greatest benefit, but it requires updating and fixing applications and carries more risk.

Whichever approach you take, get an inventory of all the software that uses the existing database. Ask the people maintaining it what issues could arise from migration. Find out if any components connect to the database in unusual ways or take advantage of undocumented features. Make the software changes and testing part of the migration plan.

Often the migration plan includes replacing current applications with cloud applications. Make sure there is a workable migration path and that personnel are trained in the new software. The initial migration shouldn’t try to change everything at once. Experience from the first applications will make the transition to other applications easier later.

Determine your storage requirements and allow for expected growth. Make sure the database instance will have enough capacity for the migration. Also make sure the migration server has enough storage for temporary data and logs, or the process could be very slow or fail.

Identify all accounts that need to be migrated. Review the privileges they have. If the old database was on-premises and not accessible from the Internet, accounts may have received more privileges than they strictly needed. With a cloud database, security considerations make it more important to observe the principle of least privilege.

If the migration requires downtime, schedule it when it will have the least impact. If the original database will be kept running till the switchover happens, the process will impact performance. Plan to run it when usage is low and there are few urgent requirements.

Have a rollback procedure in place in case things go wrong.

Setting up the target database

Amazon's DMS (Database Migration Service) is a powerful tool for migrations. It's important to know what it does and doesn't do. It doesn't convert schemas or create secondary indexes and foreign keys. If they're necessary, the people conducting the conversion need to generate them using other tools.

The cost of a DMS migration is low or even zero, unless the database is huge. Other factors, such as application updates and training, will generally be more significant. DMS does a lift-and-shift migration; additional work will be necessary if the database is going to be refactored.

Running the migration

Run a prototype migration before doing the live migration. This will help to catch any problems in the process and to find out how long the migration will take. Check that all applications will work with the migrated database. Do as much testing in advance as possible. It will save headaches and minimize the chances that a rollback will be necessary.

The AWS database migration will create a cloud database that should be identical to the source database, apart from intentional changes, during the switchover. Backing up the target should be disabled until switchover time. User accounts will be created.

Multi-megabyte data items, known as LOBs (large objects), are a special concern. Copying them increases the migration time. It could be more efficient to copy or regenerate them as a separate process.

Monitoring the migration is important for avoiding wasted time. The most likely failure mode is nothing happening. Monitoring will catch it immediately so the problem can be fixed then rather than the next morning. A migration that starts well can come to a standstill for various reasons; again, noticing it quickly allows making the necessary adjustments sooner.

Switching over

The first thing to do after the switchover is to test everything. The database needs to be checked for integrity and completeness. All applications need to be switched to the new accounts. Then they need to be checked to make sure they work with the new database. They should be able to connect and do their jobs. Some issues will likely need attention.

If anything goes seriously wrong and can't be fixed quickly, it will be necessary to roll back to the old database.

Going live

If testing shows that everything works, or if the problems are minor and won't take long to fix, the new database will immediately become the production database. If there are serious problems, it may be necessary to roll back to the old database. Rollbacks carry their own risks, so the decision needs careful consideration.

Expect some disruption, and deal with it without panic. If you've carried out the migration intelligently, any problems shouldn't be hard to deal with. Allow for a brief period of reduced productivity, knowing that afterward you'll have a better database than before.

For more information on best practices, see Amazon's recommendations for database migrations.

As an AWS-recognized partner, Agilisium can help you to set up an AWS database and get through all the phases, from assessment to migration to management and maintenance. Contact us to get started.

Competencies

aws PARTNER Advanced Tier Services

- Data & Analytics Services Competency
- Migration & Modernization Services Competency
- Immersion Day
- DevOps Services Competency
- Well-Architected Partner Program
- Microsoft Workloads Consulting Competency

SDPs

aws PARTNER Advanced Tier Services

- Amazon EMR Delivery
- Amazon RDS Delivery
- AWS Lambda Delivery
- Amazon Kinesis Delivery
- Amazon DynamoDB Delivery
- AWS Glue Delivery
- Amazon Redshift Delivery
- Amazon Quicksight Delivery
- Amazon EC2 for Windows Server Delivery
- Amazon OpenSearch Service Delivery

Top 3 Global Redshift Advocates for AWS customers

First AWS Partner across the world to achieve all AWS Data & Analytics competencies & SDPs

Boutique AWS Partner– QuickStart solutions, Chatbots, Migration accelerators, Optimization inspector etc.

aws database freedom

Preferred Partner for AWS Database freedom program for legacy DB migrations to AWS

✉ **sales@agilisium.com**

Agilisium is the fastest-growing Cloud Transformation & Data Analytics company with strong expertise in Data lake solutions, Data Warehouse Engineering, Data Migration & Modernization, Data Visualization, and Cloud Optimization services. Agilisium is an AWS Advanced Consulting Partner who helps companies architect, build, migrate, and manage their application workloads to accelerate their journey to the agile cloud, achieve desired business outcomes, and reach new emerging global markets. Learn More at www.agilisium.com.