



Zero-Downtime AWS Migration: Your A-to-Z Playbook

www.ispirer.com

Table of contents

- I. Why AWS: Quick Look 2
- II. Find Your Perfect Home in Cloud 3
- III. Pick the Right Path 5
- IV. Assess and Plan 6
- V. An Arsenal for Every Task 7
- VI. The 4-Step Migration Launch Sequence 8
- VII. Life in Cloud 9
- VIII. Contact Us 10

Ready to swap your troublesome old databases for a modern, powerful AWS cloud solution? This guide is your roadmap. We'll show you how to move to the cloud smoothly, without downtime, and unlock major benefits for your business.

Why AWS: Quick Look



Huge ROI

Companies report a 294% three-year ROI after moving to AWS.



Rock-Solid Reliability

Get up to 99.999% availability for your databases.



Smart Savings

AWS can cost as little as 1/10th of traditional commercial databases.



Scalability on Demand

AWS allows you to scale resources up or down in minutes.



Top-Notch Security

Meet strict compliance requirements like GDPR, HIPAA, and PCI DSS.



Reduced Overhead

Automate time-consuming tasks like patching, backups, and setup.

Find Your Perfect Home in Cloud

First things first: where will your data live? AWS offers purpose-built databases for every need. They fall into two main families: Relational (SQL) and Non-Relational (NoSQL).

Relational Databases for Structured Data

These are the traditional databases you know, storing data in neat tables with rows and columns. They're perfect for e-commerce, CRM, and financial systems.



Amazon RDS

↳ This is your managed service for popular databases like MySQL, PostgreSQL, Oracle, and SQL Server. [RDS](#) handles the tedious admin work — like patching, backups, and setup — so your team can focus on what matters.



Amazon Aurora

↳ Built for the cloud, [Aurora](#) is a beast. It's compatible with MySQL and PostgreSQL but way faster. It's the top choice for high-demand enterprise and SaaS applications.

✓ Up to **5x faster** than standard MySQL

✓ Up to **3x faster** than standard PostgreSQL

Non-Relational Databases for Flexibility and Scale

NoSQL databases are built for the speed and variety of modern data. They handle flexible, unstructured data with ease and are perfect for gaming, IoT, and mobile apps.



Amazon DynamoDB

↳ A key-value and document database that delivers single-digit millisecond speed at any scale. It's serverless, meaning it scales automatically with zero administration.



Amazon ElastiCache

↳ An in-memory cache that provides microsecond latency. Use it for real-time applications, session stores, and leaderboards to make everything feel instant.



Amazon DocumentDB

↳ If you use MongoDB, this is your easy path to the cloud. It's a fast, scalable, and highly available service that's fully MongoDB-compatible.



Amazon Neptune

↳ A graph database made to understand relationships in data. It's the brains behind recommendation engines, fraud detection, and social networks.

AWS Database Services

AWS service	Database type	Common use cases	Key performance characteristic
Amazon RDS	Relational	Web/mobile apps, e-commerce, CRM, financial systems	Managed service for popular database engines
Amazon Aurora	Relational	High-throughput enterprise apps, SaaS products	Up to 5x faster than MySQL, 3x faster than PostgreSQL
Amazon DynamoDB	Key-value & document	Gaming, ad tech, IoT, serverless applications	Single-digit millisecond latency at any scale
Amazon DocumentDB	Document	Content management, catalogs, migrating MongoDB workloads	MongoDB-compatible, highly available
Amazon ElastiCache	In-memory	Caching, session stores, real-time leaderboards	Microsecond latency
Amazon Neptune	Graph	Recommendation engines, fraud detection, social networking	Optimized for highly connected datasets



The power of the cloud is using purpose-built databases. Don't just move your old SQL database because it's what you know. Consider if a NoSQL option like DynamoDB could unlock better performance and scalability for your modern applications.

Pick the Right Path

Once you know your destination, you need to choose how you'll get there. The 6 R's framework helps you pick the right strategy.



Rehost (Lift and Shift)

The fastest route. You move your existing database server as-is to AWS. It's quick and easy, but you get fewer cloud benefits.



Replatform (Move and Improve)

A popular middle ground. You move your database to a managed service like [Amazon RDS](#). You offload management to AWS but keep your core database engine the same. It's a great balance of effort and reward.



Refactor / Re-architect

The most transformative path. You redesign your application and database to be fully cloud-native, like moving from [Oracle to Amazon Aurora](#). It's the most work but delivers the biggest long-term gains in performance and cost savings.



Repurchase (Drop and Shop)

You move to a different product entirely, like switching from a self-hosted CRM to a SaaS solution.



Retire

You identify and decommission applications and databases you no longer need, saving money and effort.



Retain

Some things aren't ready to move. You choose to keep them on-premises for now.

A Phased Approach is a Smart Approach

Refactoring an entire application during the initial migration can be risky.

Best practice: Replatform first to get into the AWS ecosystem quickly and safely. Once you're there, you can refactor in a second, lower-risk phase.

Assess and Plan

This is the most critical phase. Rushing here leads to surprises and budget overruns. A solid plan is built on solid data.

1 Define Your Scope

This is the most critical phase. Rushing here leads to surprises and budget overruns. A solid plan is built on solid data.



Turn to [InsightWays](#) and automatically scan your old database to quantify its size and complexity. It catalogs every object, table, and line of code, giving you a complete inventory.

2 Uncover Potential Issues

With a deep analysis, you can spot conversion challenges and incompatibilities between your old database and your new AWS target before they become problems.

3 Estimate Cost and Duration

The assessment provides a detailed report on your database. Armed with this data-backed report, you can accurately forecast the time, effort, and investment your migration will require.



Read more: [New InsightWays Reports: Clearer Insights, Better Plans](#)

An Arsenal for Every Task

A successful migration requires the right tools not only for planning, but for actual migration and further optimization. AWS and its partners offer a powerful suite to handle every stage of the journey.

Relational Databases for Structured Data



Ispirer InsightWays

Scan your source databases and get a detailed report on complexity, dependencies, and potential issues **for free**.



AWS Migration Hub

Your command center. It provides a single dashboard to track the progress of your migration across various AWS and partner tools.



AWS Application Discovery Service

Collects the technical data about your on-premise servers, their performance, and network dependencies to inform your planning.

Database Migration & Modernization Tools



Ispirer SQLWays

For complex migrations, SCT leaves much manual work, while SQLWays automates up to 99% of complex business logic, reducing time and risk.



AWS Database Migration Service

DMS handles the actual data replication from your source to the target, with features that allow for near-zero downtime.



AWS Schema Conversion Tool

SCT automatically converts your source database schema and some code to a format compatible with your new database engine.

Server & Large-Scale Data Transfer Tools



AWS Application Migration Service

The primary tool for lift and shift (Rehosting). It replicates entire servers to AWS with minimal disruption.



AWS DataSync

Accelerates online data transfer between on-premise storage and AWS, moving data up to 10x faster than standard tools.



AWS Snowball

For when the internet isn't fast enough. AWS ships you a physical device to load petabytes of data and ship it back for upload to the cloud.



Read more: [20 Best AWS Tools for Cloud Migration](#)

The 4-Step Migration Launch Sequence

With your plan in place, it's time for the technical execution — where the magic happens.

1 Schema & Code Conversion

This means translating the structure (schema) and the logic (code) of your database.



For complex migrations, especially moving from one database type to another (e.g., Oracle to PostgreSQL), you need a powerful translator.

[SQLWays](#) automates the conversion of your schema, tables, views, and — most importantly — the complex business logic in your stored procedures, functions, and triggers.

- ✓ Achieve up to **99% automation** on code conversion
- ✓ Deliver your project **2-3 times faster** than with manual efforts



Read more: [DB2 to Aurora Migration Guide](#)

2 Review and Refine

Your team, alone or with help from Ispirer's experts, will validate the converted code to ensure it meets your standards and will perform perfectly in the new AWS environment.

3 Test in a Controlled Environment

Before going live, run functional and performance tests. This confirms that everything works flawlessly and ensures a predictable and smooth transition for your users.

4 Data Migration and Cutover

Data Migration

The actual data is moved from your source database to your new AWS database. AWS Database Migration Service or [SQLWays](#) can achieve this with near-zero downtime, keeping your apps running throughout the process.

Final Cutover

Once the data is fully synced, you flip the switch. Your applications are redirected to the new Amazon database. Your users immediately start to benefit from the speed and reliability of AWS.

Life in Cloud

You've made it! But the journey isn't over. Now you can focus on getting the most value out of the cloud.



Validate Performance

After the cutover, carefully confirm that all your applications are running smoothly and performing optimally to ensure stable operations.



Strengthen Security

Use AWS tools to lock down access, encrypt data, and ensure you consistently meet all compliance requirements.



Set Up Backups

Implement a solid backup and disaster recovery plan by leveraging services like AWS Backup to keep your systems secure and resilient.



Optimize Costs

Use tools like AWS CloudWatch to monitor performance and right-size your database, ensuring you only pay for what you need.

***You are now officially modernized
on the Amazon cloud platform!***



Let's Plan Your AWS Migration Project Together

Feeling ready to make the move? We're here to help you figure out the resources and tools you need to land successfully on the AWS cloud.

Book a call with an Ispirer Expert today!

Contact us →