Migrate Legacy Systems to Modern Application Development Environments with MongoDB Atlas on AWS

by Paresh Saraf, Igor Alekseev, Geoff Bernard, and Peter Kim | on 17 NOV 2021 | in AWS Partner Network, Customer Solutions, Database, Intermediate (200), Mainframe & Legacy, Migration, Thought Leadership | Permalink | Comments | Share

By Paresh Saraf, Partner Solutions Architect – MongoDB
Geoff Bernard, Partner Solutions Architect – MongoDB
Peter Kim, Partner Solutions Architect – MongoDB
Igor Alekseev, Partner Solutions Architect – AWS

Cloud computing represents a once-in-a-generation shift in how businesses operate. Elastic, on-demand computing allows organizations to provision compute resources as needed instead of unnecessarily maintaining a large overhead of infrastructure for occasional demand spikes.

Cloud providers offer a vast catalog of software and services for developers to quickly build new features and applications. This allows companies to deploy revenue-generating features more rapidly.

To reap the benefits of cloud computing—including increased agility, just-in-time provisioning of resources, and accelerated delivery of new features to market—organizations cannot simply take their existing applications and architectures in their legacy on-premises environment and move them as-is to the cloud.

This type of cloud migration is commonly referred to as "lift and shift" and organizations that use this approach miss out on what's made cloud computing so transformative for the innovative companies that have fully embraced it.

True cloud modernization enables you to move data and simultaneously apply the latest innovations in development methodologies, architectural patterns, and technologies to refresh your portfolio of existing applications.

In this post, we describe solutions that companies use to modernize by migrating their data from legacy databases to **MongoDB Atlas**—a leading general-purpose, document-based platform—on Amazon Web Services (AWS). MongoDB is an AWS Partner with the Data & Analytics Competency that helps AWS customers around the world rapidly innovate and accelerate vital cloud modernization projects.

True Cloud Modernization Requires More Than a "Lift and Shift"

Lift and shift is limited in its ability to transform businesses because it continues to rely on outdated, legacy technologies and architectures that limit flexibility and slow productivity.

Applications built as monoliths cannot benefit from the cloud's ability to scale out dynamically. This is because functional components are tightly coupled and unable to scale independently and according to function. Monolithic applications also generally have long dev/test/release cycles that don't benefit from the cloud's ability to provision and deploy resources as rapidly as needed.

In contrast, in a microservices architecture, the discrete functions can be developed and released by independent development teams, allowing them to work rapidly and without dependencies. Microservices can be scaled independently, making the architecture more cost efficient and responsive to increased demand.

Legacy relational databases are also limited in their ability to fully achieve the agility and scalability of a document model database like MongoDB. Legacy relational databases were not built to scale horizontally. Sharding data to handle large data volumes and ensure lower latency is typically a significant manual effort that requires custom application logic to query across multiple shards and aggregate results.

In contrast, document databases like MongoDB are built to be distributed and to handle sharding transparently to the developer. With MongoDB, no application code changes are necessary when an application has to scale out from a 10MB to a 500TB dataset.

Additionally, rigid relational schemas often require downtime and significant application code updates to make even simple modifications like adding a new data attribute. In MongoDB, entities are modeled as documents that map naturally to the same objects that developers are used to working with in their programming languages. This removes the need for a complex object-relational mapping layer, and allows developers to build and release new features quicker.

Replacing Your Relational Database with MongoDB

At the solution architecture and engineering level, the need to modernize means meeting the demands of the business at scale, at speed, and within budget. It means reimagining the architectures and technologies that underlie applications. As a document-based database, MongoDB is an excellent choice to get users started on their journey.

As an assumed storage layer, relational databases inhibit the ability to efficiently scale, add more complexity to modelling, and slow down developer productivity. Alternatively, the flexibility and scalability of a document-based database like MongoDB is perfectly suited for a microservices architecture.

Whether organizations are dealing with the migration of a specific application, full mainframe offload, or vendor lock-in scenario, MongoDB on AWS is the more productive and cost-effective choice.

Below, we observe some common patterns for how MongoDB on AWS is used. For a more general treatment of other patterns, check out MongoDB use cases.

- Monolith to microservice – With its flexible schema and capabilities for redundancy, automation, and scalability, MongoDB (and MongoDB Atlas, its managed services version) is well suited for microservices architecture. Together, MongoDB Atlas and microservices on AWS can help organizations better align teams, innovate faster, and meet today's demanding development and delivery challenges with full sharding across regions and globally.
- Legacy modernization – Relational databases impose a tax on the business—specifically, a data and innovation recurring tax (DIRT). By modernizing with MongoDB, you can build new business functionality 3-5x faster, scale to millions of users wherever they are on the planet, and cut costs by 70% and more—all by unshackling yourself from legacy systems and, at the same time, taking advantage of the AWS environment.
- Mainframe offload – MongoDB can help offload key applications from the mainframe to a modern data platform without impacting core systems, while also helping you attain agility and reduce costs.
- Real-time analytics – MongoDB makes it easier to scale to the needs of real-time analytics with Atlas on AWS. Coupled with AWS analytics, such as Amazon Redshift, you can quickly run complex queries and glean rich, actionable insights.
- Mobile application development – MongoDB Realm runs on AWS and helps companies build better apps faster with edge-to-cloud sync and fully managed backend services including triggers, functions, and GraphQL.

Resources

Why Work with AWS Partners
Training for Partners
AWS Competency Partners
Managed Service Providers (MSPs)
Partner Central Login
Case Studies and References
AWS Sponsorship Opportunities

Follow

AWS Partners
AWS Cloud
APN LinkedIn
APN YouTube
RSS Feed
APN Email Updates

Although the move to a document-based database like MongoDB may seem like an ambitious undertaking for those who have spent a lifetime working in relational databases, the amount of time and cost savings provided, combined with the opportunity for innovation, makes document-based databases the undeniable choice of the future.

MongoDB Atlas on AWS: A Reference Architecture for Modernization

When you decide to modernize, the first step is to move your data to MongoDB's document database. There are two ways to do this: through a one-time data migration, or an ongoing real-time data synchronization.

Either way, a transform layer must be built in a format that makes sense for APIs and enables you to model your data in a collection. AWS tools help faciitate both migration options.

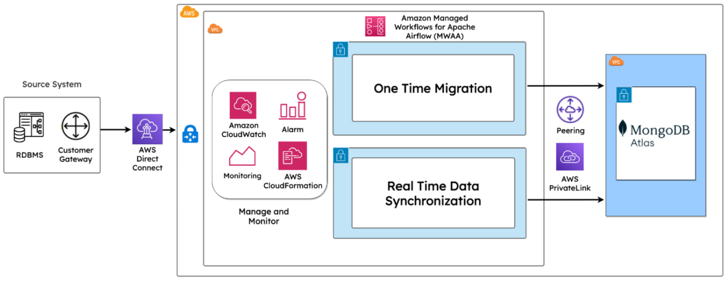


Figure 1 – Solution architecture.

A one-time data migration involves an initial bulk extract, transform, load (ETL) of data from the source relational database to MongoDB.

You can use AWS Database Migration Service (AWS DMS), Apache Sqoop, or Spark SQL's JDBC connector to extract data from the source and store it in Amazon Simple Storage Service (Amazon S3) temporarily. AWS Glue or custom Spark jobs can then transform the data and load into MongoDB Atlas.

MongoDB has a native Apache Spark connector which can store Spark DataFrames as collections.

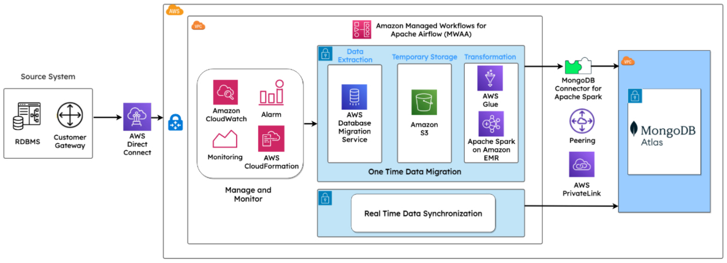


Figure 2 – One-time data migration.

In most migrations, the source database will not be retired for a few weeks to months. In such cases, MongoDB Atlas needs to be kept up to date with the source database through real-time data synchronization.

Change data capture (CDC) tools like AWS DMS CDC, Debezium, or Qlik Replicate are used to capture the changes, which can then be pushed to message queues like Confluent or Amazon Managed Streaming for Apache Kafka (Amazon MSK).

Users can write custom transformation jobs using Java or Python which can consume the data from the message queue, transform it, and push it to MongoDB Atlas using native drivers.

AWS CloudFormation can be used to provision all the components, while Amazon CloudWatch handles monitoring. Amazon Managed Workflows for Apache Airflow (MWAA) will help with orchestrating all the workflows.

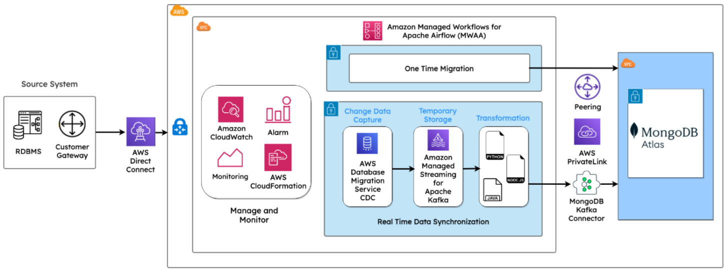


Figure 3 – Real-time data synchronization.

Why Developers Prefer Modernizing with MongoDB

MongoDB has become a sought-after database for building out APIs for modern application development, according to Stack Overflow. Here are some of the reasons:

- Developers can get more done with JSON—it's just easier to work with JSON, as well as objects in driver languages.
- Developers and analysts can have a common, human-readable, natural language. They can get further without having to decrypt the data layer, and they can trust the data is optimized for sub-second and sub-millisecond reads and writes.
- MongoDB's key competitive advantage is its document model. The API can effortlessly query into any layer of a document in an intelligible and efficient manner with minimal chatter across the wire.
- The pains of figuring out the assumed cost of the classic SQL-based JOIN have been taken out of the equation, but MongoDB still guarantees ACID compliance. In addition, MongoDB provides ACID guarantees across multiple operations, databases, documents, and servers in a cluster. MongoDB is designed to be strongly consistent by default but also tunable depending on the application's requirements.
- MongoDB can effortlessly place data where it needs to be with cross-regional sharding and replication. Sharding can be done by geography, hash, or range in order to reduce latency or scale horizontally. Real-time analytics and read-only workloads can be performed on up to 50 replicas of the data and at the desired consistency level.
- MongoDB Atlas makes it easy to spin up new clusters, scale according to demand with minimal effort, and simplify work efforts and workloads. With Atlas, you can easily enable full-text search on a field, use Data Lake and Online Archive to query data in Amazon S3, and run a real-time analytics layer without having to manage multiple

solutions.
- MongoDB, as a company, has partnered with multiple solution-based organizations in order to provide connectors for a broader environment. If there's a need to analyze data inside a business intelligence (BI) reporting tool, within Spark, or even through notification by pub/sub for event-driven systems, all of these mechanisms can be set up—on premises, with a cross-cloud provider, or with a hybrid approach.

In addition to the core database requirements, MongoDB has listened to their user community and built out a broad portfolio of data platform tools to accelerate developer productivity and open up new use cases for extracting value from their data in MongoDB.

Here are some of the newer products and capabilities:

- MongoDB Realm – Suite of tools and services to accelerate mobile and web application development with features like edge-to-cloud sync and fully-managed GraphQL APIs.
- Atlas Search – Lucene-based full text search API with capabilities such as auto-complete and boosting.
- Atlas Data Lake – Integration with Amazon S3 enabling MongoDB to query cold storage and optionally federate queries across data stored in both S3 and MongoDB.
- Atlas Charts – Data visualization suite that can generate charts to embed in developers' applications securely.

These capabilities greatly simplify matters for IT shops looking to modernize their applications. Now, let's look at MongoDB Atlas on AWS specifically.

Why MongoDB Atlas on AWS?

MongoDB Atlas is well integrated into the AWS environment, as shown in *Figure 4*. The document-based database works seamlessly with AWS products. To learn more about the common integration and project requirements, see Managed MongoDB on AWS.

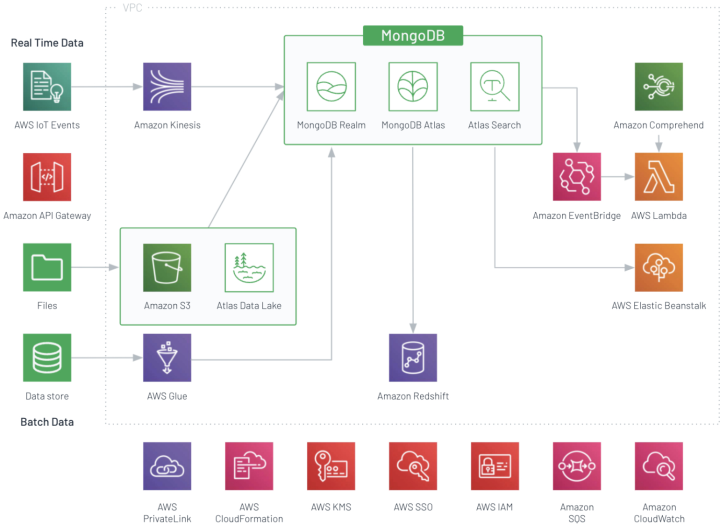


Figure 4 – MongoDB and the AWS environment.

MongoDB Atlas is a fully managed version of MongoDB as a service. Atlas provides three key benefits:

1. Significant reduction in the operational effort of running MongoDB yourself.
2. Out-of-the-box security and resilience.
3. Services and tools that extend the core MongoDB features to improve developer productivity.

Atlas significantly reduces the day-to-day work required for maintenance tasks like monitoring and backups. Backups in Atlas are automated by default, achieving a recovery point objective (RPO) down to one minute. Atlas offers monitoring of 100+ metrics at the database and system levels in the admin user interface (UI) in addition to providing alerting and integration with third party observability tools like DataDog.

MongoDB Atlas ensures your MongoDB cluster is secure and resilient from the start. Setting up a highly available cluster with automatic failover, even distributed globally and across multiple cloud providers, is as simple as making a few selections in the Atlas UI or an Atlas API call.

Atlas creates clusters with encrypted storage and encrypted transit by default. Atlas also supports bringing your own encryption key managed by your preferred key management service like AWS Key Management Service (AWS KMS).

Finally, MongoDB Atlas offers services and tools beyond the core database that developers typically need in their applications.

Instead of incorporating numerous other technologies into the architecture and designing and maintaining integrations to those products, Atlas offers many of these services along with your core database cluster.

For example, instead of standing up a separate search engine and creating migration jobs to sync data between the database and search engine, MongoDB Atlas provides a Lucene-based, full-text search engine queryable through the same MongoDB Query Language (MQL) used to query the database.

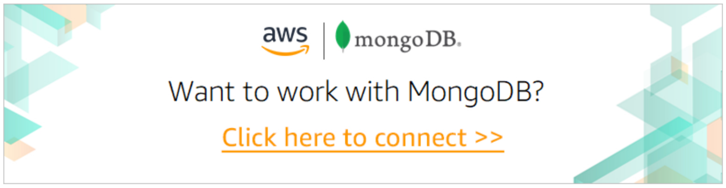
MongoDB Atlas provides significant value as a fully-hosted and managed database-as-a-service to offload the burden of operations, maintenance, and security to the world's leading MongoDB experts. It also offers services like Atlas Search, Realm, Atlas Data Lake, and more to make MongoDB Atlas the most comprehensive data platform in the market.

Conclusion

Together, AWS and MongoDB help customers reap the full benefits of cloud computing, including increased agility, just-in-time provisioning of resources, and acceleration of release cycles. Move beyond a "lift and shift" approach and migrate enterprise-level mainframe applications to new, leading-edge architectures and databases today.

To learn more about how MongoDB Atlas is integrated with AWS, see the following resources:

- MongoDB Atlas on AWS Quick Start
- Atlas Data Lake
- Realm + EventBridge How-to
- Kinesis Data Firehose integration
- PrivateLink
- MongoDB Atlas on AWS



MongoDB – AWS Partner Spotlight

MongoDB is an AWS Competency Partner. Their modern, general purpose database platform is designed to unleash the power of software and data for developers and the applications they build.

Contact MongoDB | Partner Overview | AWS Marketplace

***Already worked with MongoDB?** Rate the Partner

**To review an AWS Partner, you must be a customer that has worked with them directly on a project.*

TAGS: Amazon CloudWatch, Amazon MSK, Amazon Redshift, Amazon S3, APN Partner Guest Post, APN Partner Spotlight, APN Partner Success Stories, APN Technology Partners, Application Modernization, AWS CloudFormation, AWS Competency Program, AWS DMS, AWS Glue, AWS Key Management Service, AWS KMS, AWS Marketplace, AWS Partner Solutions Architects (SA), AWS Public Sector Partners, AWS Service Delivery Program, Database Migration, Database Services, Legacy Modernization, Migration, MongoDB, MongoDB Atlas, The Next Smart