Application Modernization with MongoDB Atlas on AWS

by Abhinav Mehla and Igor Alekseev | on 17 SEP 2021 | in AWS Partner Network, Customer Solutions, Database, Intermediate (200), Thought Leadership | Permalink | Comments | Share

By Abhinav Mehla, Sr. Director, Cloud Partners – MongoDB
Igor Alekseev, Sr. Partner Solutions Architect – AWS

In this post, we'll explore the importance of modernizing your data infrastructure to speed up application development and iteration, as well as key features and benefits of using MongoDB Atlas on AWS to get the job done. Additionally, we'll review a case study of one organization's modernization journey.

Competitive advantage often lies in a company's ability to create modern customer experiences. But for many, an unwieldy data infrastructure can get in the way and prevent innovation.

To build and evolve applications quickly, gain timely access to key insights, and ultimately stay ahead of the pack, many companies will need to modernize their data infrastructure.

Most modernization projects begin with leaving (at least part of) the past behind. Many business applications have traditionally been built on legacy systems based on relational database management systems (RDBMS), which store data in a rigid, table-based (or tabular) format. Unfortunately, these rigid data structures are hard to change and thus slow down application development.

Tabular data models also aren't intuitive and don't match the way most developers think and code at the application layer: a single logical entity, like a customer, may be spread across dozens of tables.

Furthermore, most relational databases don't have a distributed architecture built in. Without automatic failover, an application's downtime risk is increased, and without horizontal scaling, the data layer is expensive and complicated to grow with a global customer base. The consequences of an unwieldy data infrastructure are manifold.

Developer productivity is inhibited as they spend time trying to innovate on top of a data model that doesn't match application logic and is hard to evolve. Administering databases is onerous, especially if you need to build in your own resilience and scaling—this is even more time-consuming if you're managing your own infrastructure on-premises.

Some organizations will attempt to add additional data systems alongside their RDBMS for new capabilities, only to encounter new problems around data integration and movement, data duplication, and complex architectures, resulting in a fragmented development experience. The net effect of a poor, out-of-date data infrastructure is that innovation slows to a crawl.

The solution? Adopting a global cloud database service that can help you build, run, and improve your modern applications.

MongoDB Atlas on Amazon Web Services (AWS) lets you build applications that are highly available, performant at a global scale, and compliant with the most demanding security and privacy standards. When you use MongoDB Atlas on AWS, you can focus on driving innovation and business value, instead of managing infrastructure.

Why Modernize with MongoDB Atlas on AWS?

The combined capabilities of MongoDB Atlas and AWS make it easy to focus on consistently shipping exceptional features and products anywhere in the world, all while improving developer productivity.

MongoDB's flexible document model makes building and evolving applications simple, replacing the rigidity and development friction of relational databases. Intelligent performance advice allows users to view index suggestions based on slow running queries, and then build them with a few clicks.

You can rely on Atlas and AWS to build modern, data-driven applications and to scale them, optimize their performance, and simplify their data layer.

MongoDB Atlas's unified data architecture also helps you reduce complexity across systems. Interact with a single API to meet all of your core application data needs, from core transactional applications to built-in search, real-time analytics to data lake workloads, and more.

Extend your applications with integrations between Atlas and AWS services for advanced capabilities and simple connection with your AWS application tier. You can also build event-driven applications with Amazon EventBridge or connect with Amazon Kinesis to load streaming data, enable real-time enrichment, and more.

With MongoDB Atlas and AWS, you can deploy confidently anywhere in the world: Atlas is available in 18 AWS Regions globally.

With distributed fault tolerance designed for resilience, Atlas offers industry-leading uptime SLA of 99.995%. Combined with built-in backups and compliance standards, you can have confidence using Atlas on AWS for your most sensitive data and your most important workloads.

Softinstigate: Winning the Fast-paced Italian Marketplace

Softinstigate is a highly specialized services company focused on providing cutting-edge engineering solutions for clients who want to accelerate their software development and digital transformation.

With deep expertise in enterprise data products, CEO Andrea Di Cesare and Co-Founder Maurizio Turatti have always understood the critical role that an agile database plays in delivering such success. When they found themselves spending too much time on technical fixes to keep servers and databases running instead of solving their clients' business problems, they knew they had to make a change.

Eager to accelerate development cycles, meet demand, and win contracts in their home country of Italy, where many small businesses are able to move at lightning speed, Andrea and Maurizio chose to leave the inflexibility of relational databases behind in order to overcome its limitations and succeed in a new paradigm.

With the rapid ascension of web and mobile apps, the Softinstigate founders knew they could easily manage data from such sources in a document database. They made the decision to build a Java app server on top of MongoDB.

RESTHeart: A Ready-to-use REST API for MongoDB

Softinstigate developed RESTHeart, a ready-to-use REST API for MongoDB, and the company now deploys the majority of its products and services on MongoDB Atlas running on AWS.

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

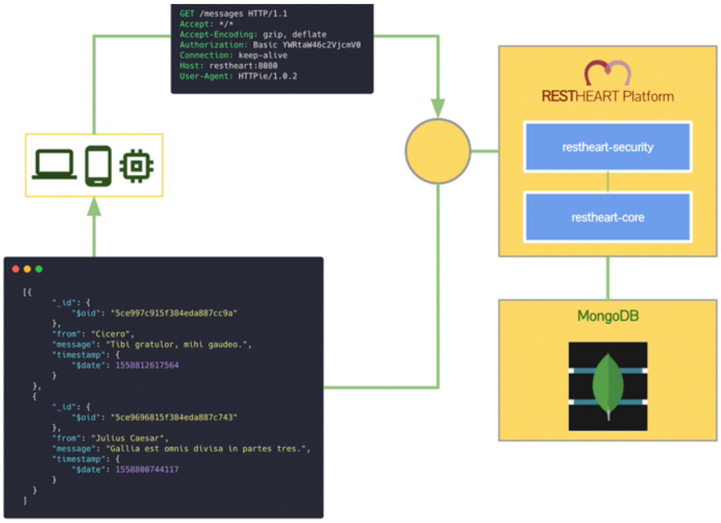
RESTHeart is a stateless REST API microservice that uses simple HTTP calls to unleash the power of MongoDB. It is written in Java and requires no server-side coding, thereby freeing developers to focus on developing innovative, value-adding apps.

Many of Softinstigate's customers are running RESTHeart on Amazon Elastic Container Service (Amazon ECS), as a simpler alternative to managing a Kubernetes cluster. The architecture also includes a Multi-Availability Zone Application Load Balancer on top of the ECS cluster to get automatic load balancing and blue/green hot deployments.

Together with MongoDB Atlas always on and globally available clusters, RESTHeart's architecture removes any single point of failure on the platform.

Softinstigate's RESTful design strategy was to model sessions and transactions as first-class REST resources. In MongoDB, an operation on a single document is always atomic, and that's one of the reasons Softinstigate opted to model data using embedded documents instead of creating relationships between multiple documents and collections.

Softinstigate gained more advantages with the launch of MongoDB 4.0, including atomicity and strong consistency across databases, collections, documents, and shards using transactions.



```
GET /messages HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: Basic YWRtaW46c2VjcmV0
Connection: keep-alive
Host: restheart:8080
User-Agent: HTTPie/1.0.2
```

```
[{
        "_id": {
            "$oid": "5ce997c915f384eda887cc9a"
        },
        "from": "Cicero",
        "message": "Tibi gratulor, mihi gaudeo.",
        "timestamp": {
            "$date": 1558812617564
        }
    },
    {
        "_id": {
            "$oid": "5ce9696815f384eda887c743"
        },
        "from": "Julius Caesar",
        "message": "Gallia est omnis divisa in partes tres.",
        "timestamp": {
            "$date": 1558800744117
        }
    }
]
```

Figure 1 – RESTHeart's simple, MongoDB-driven architecture.

Virtually, any device can send an HTTP request to MongoDB through the API. RESTHeart sends queries to MongoDB using the Java driver and sends back the HTTP response as a JSON message.

Softinstigate is quick to point out that in addition to JSON, RESTHeart can accommodate virtually any type of content.

> "If you want to have replication with other databases it is a much larger effort. Failover and upgrade are easy with MongoDB; to scale a single node up and down, it's a click of the mouse. We take this level of convenience for granted with MongoDB." – *Co-Founder Maurizio Turatti*

For instance, one Softinstigate customer experiences extreme peak loads with its data traffic, and MongoDB's Autoscale feature scales up to handle them.

In addition to the standard REST API, starting from version 6 RESTHart also offers complete GraphQL and Websocket APIs on top of MongoDB.

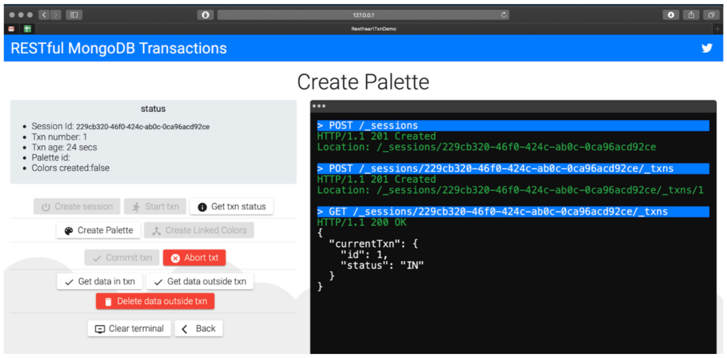
Use Case: Executing RESTful Multi-document Transactions

The following simple use case shows the power of RESTful multi-document transactions to execute transactions using very little code.

The client uses an Angular app that interacts with the server through simple HTTP requests. The server stack comprises MongoDB and RESTHeart.

RESTHeart exposes the REST API, which meets all the client's needs to read and write data, both normally and transactionally. All of this occurs with no server-side code. The backend is a standard installation of RESTHeart that provides the API out of the box.

Figure 2 is a user screen view in RESTHeart. On the left side are buttons to control the execution of the transaction step-by-step. On the right, you can see request logs showing interactions between the client and server describing each REST request.



- Session Id: 229cb320-46f0-424c-ab0c-0ca96acd92ce
- Txn number: 1
- Txn age: 24 secs
- Palette id:
- Colors created:false

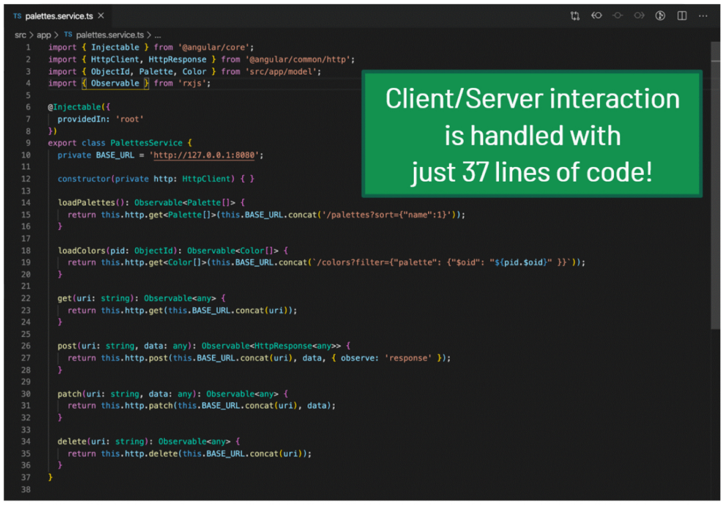
```
> POST /_sessions
HTTP/1.1 201 Created
Location: /_sessions/229cb320-46f0-424c-ab0c-0ca96acd92ce

> POST /_sessions/229cb320-46f0-424c-ab0c-0ca96acd92ce/_txns
HTTP/1.1 201 Created
Location: /_sessions/229cb320-46f0-424c-ab0c-0ca96acd92ce/_txns/1

> GET /_sessions/229cb320-46f0-424c-ab0c-0ca96acd92ce/_txns
HTTP/1.1 200 OK
{
  "currentTxn": {
    "id": 1,
    "status": "IN"
  }
}
```

Figure 2 – User screen view in RESTHeart.

For this use case, data is stored in two collections: Palettes and Colors. Each Palette is composed of five Colors and is implemented by a property in Colors that references the corresponding palette.

Creation of the Palette requires writing six documents across the two collections, and the application writes them through a multi-document transaction to enforce data consistency.

As shown in *Figure 3*, creating this app required only 37 lines of code.



```
import { Injectable } from '@angular/core';
import { HttpClient, HttpResponse } from '@angular/common/http';
import { ObjectId, Palette, Color } from 'src/app/model';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class PalettesService {
  private BASE_URL = 'http://127.0.0.1:8080';

  constructor(private http: HttpClient) { }

  loadPalettes(): Observable<Palette[]> {
    return this.http.get<Palette[]>(this.BASE_URL.concat('/palettes?sort={"name":1}'));
  }

  loadColors(pid: ObjectId): Observable<Color[]> {
    return this.http.get<Color[]>(this.BASE_URL.concat(`/colors?filter={"palette": {"$oid": "${pid.$oid}" }}`));
  }

  get(uri: string): Observable<any> {
    return this.http.get(this.BASE_URL.concat(uri));
  }

  post(uri: string, data: any): Observable<HttpResponse<any>> {
    return this.http.post(this.BASE_URL.concat(uri), data, { observe: 'response' });
  }

  patch(uri: string, data: any): Observable<any> {
    return this.http.patch(this.BASE_URL.concat(uri), data);
  }

  delete(uri: string): Observable<any> {
    return this.http.delete(this.BASE_URL.concat(uri));
  }
}
```

Client/Server interaction
is handled with
just 37 lines of code!

Figure 3 – Palettes of Colors app code.

Today, all Softinstigate customer deployments are on MongoDB-driven RESTHeart running in the cloud on AWS.

The company is now leveraging the capabilities of MongoDB Atlas to grow and maintain its existing revenue streams while expanding up-market. They haven't touched a relational database in six years.

Highlights include:

- Nearly 100% availability supporting Softinsitgate's dual go-to-market strategy.
- Advantages in agility and resource savings.
- Accelerates clients' time to market by up to 20x.
- Simple MongoDB-AWS architecture creates trusted relationships, especially in Italy's fast-moving marketplace.
- 100m+ downloads.

To learn more, read the full case study.

Conclusion

Modernizing your data infrastructure isn't just a nice-to-have; it's an industry-agnostic requirement. Your ability to innovate depends on it, and MongoDB Atlas on AWS can help.

Together, we can help you—and your clients— accelerate application development cycles, scale faster, reduce costs, and deploy anywhere in the world. Reach out today to learn more.

Learn more about 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 ECS, Amazon EventBridge, Amazon Kinesis, APN Partner Guest Post, APN Partner Spotlight, APN Partner Success, APN Partner Success Stories, APN References, APN Technology Partners, Application Modernization, AWS Competency Program, AWS Partner Solutions Architects (SA), AWS Service Ready, MongoDB, MongoDB Atlas, Softinstigate, The Next Smart

View Comments

AWS PartnerCast
Attend free interactive webinars for AWS Partners hosted by AWS experts.
Learn more »

AWS Partner Network
Find an APN member to support your cloud business needs
Learn more »

AWS Training & Certifications
Free digital courses to help you develop your skills
Learn more »