



Architecture Diagrams

Migration from MongoDB to Amazon DocumentDB



Migration from MongoDB to Amazon DocumentDB: Architecture Diagrams

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Offline Approach **1**

Migration from MongoDB to Amazon DocumentDB: Offline Approach 1

Further reading 2

Diagram history 2

Online Approach **3**

Migration from MongoDB to Amazon DocumentDB: Online Approach 3

Hybrid Approach **5**

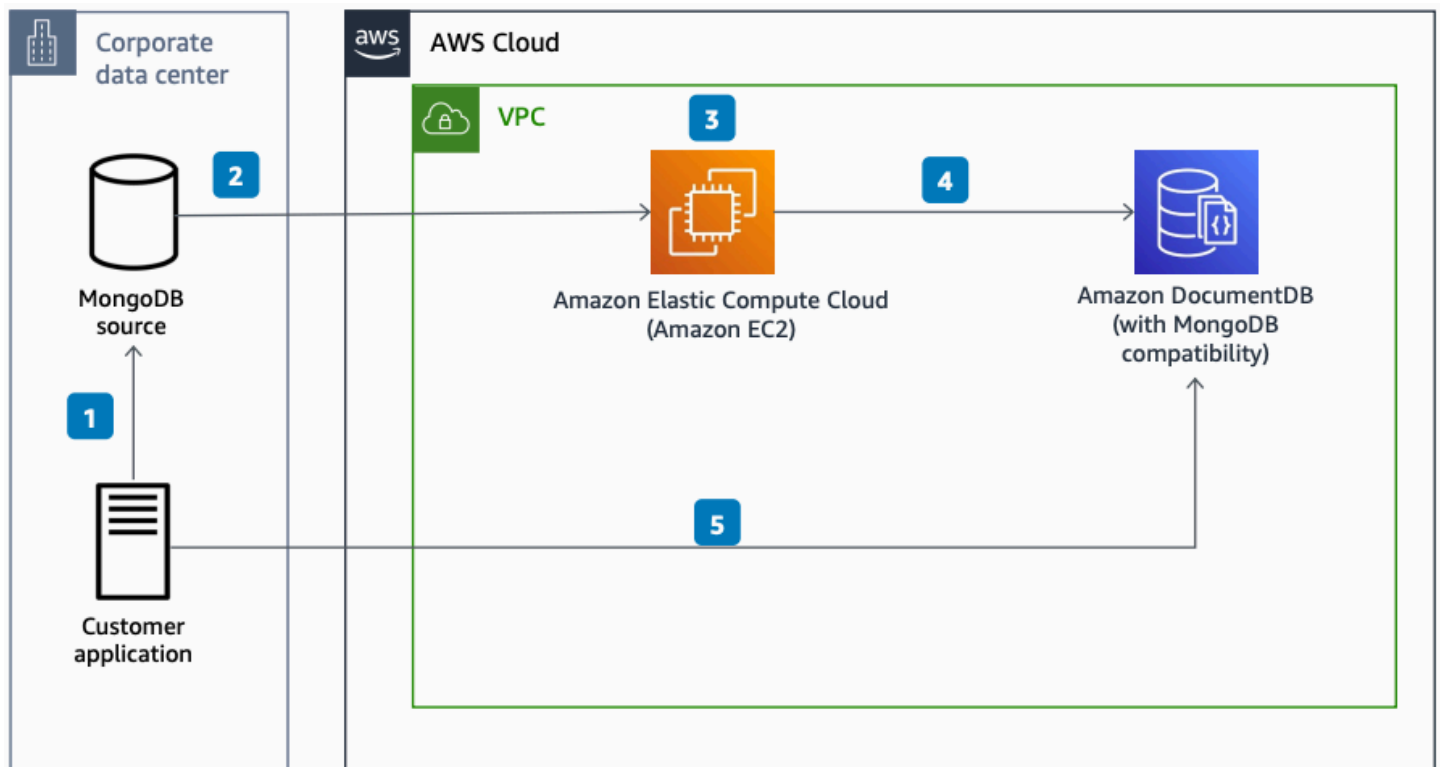
Migration from MongoDB to Amazon DocumentDB: Hybrid Approach 5

Migration from MongoDB to Amazon DocumentDB: Offline

Publication date: **March 25, 2022** ([Diagram history](#))

This architecture shows how to migrate from MongoDB to [Amazon DocumentDB](#) using the offline approach with MongoDB utilities. This is the simplest approach, but it can result in more downtime during the migration.

Migration from MongoDB to Amazon DocumentDB: Offline Approach



The following steps describe the architecture:

1. Before migration to the Amazon DocumentDB cluster, ensure that the MongoDB source and application drivers use version 3.6 or higher. Stop application writes to the source MongoDB to ensure data consistency.
2. Establish connectivity from the Amazon EC2 instance to your MongoDB source, then export the data and indexes to the Amazon EC2 migration instance using the `mongodump` tool. To reduce

the impact on the source deployment, use the MongoDB read preference option to force the dump to connect to a secondary replica set member.

3. Use the Amazon DocumentDB Index Tool to check the dumped indexes for compatibility and to pre-create the indexes on the target Amazon DocumentDB cluster. This improves overall restore time.
4. Restore exported data to your target Amazon DocumentDB cluster using the mongorestore tool. To parallelize imports, use the mongorestore numInsertionWorkersPerCollection option.
5. After data restore is complete, switch your application's database connection string to use your new target Amazon DocumentDB cluster.

Further reading

For additional information, refer to the following resources:

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	March 25, 2022

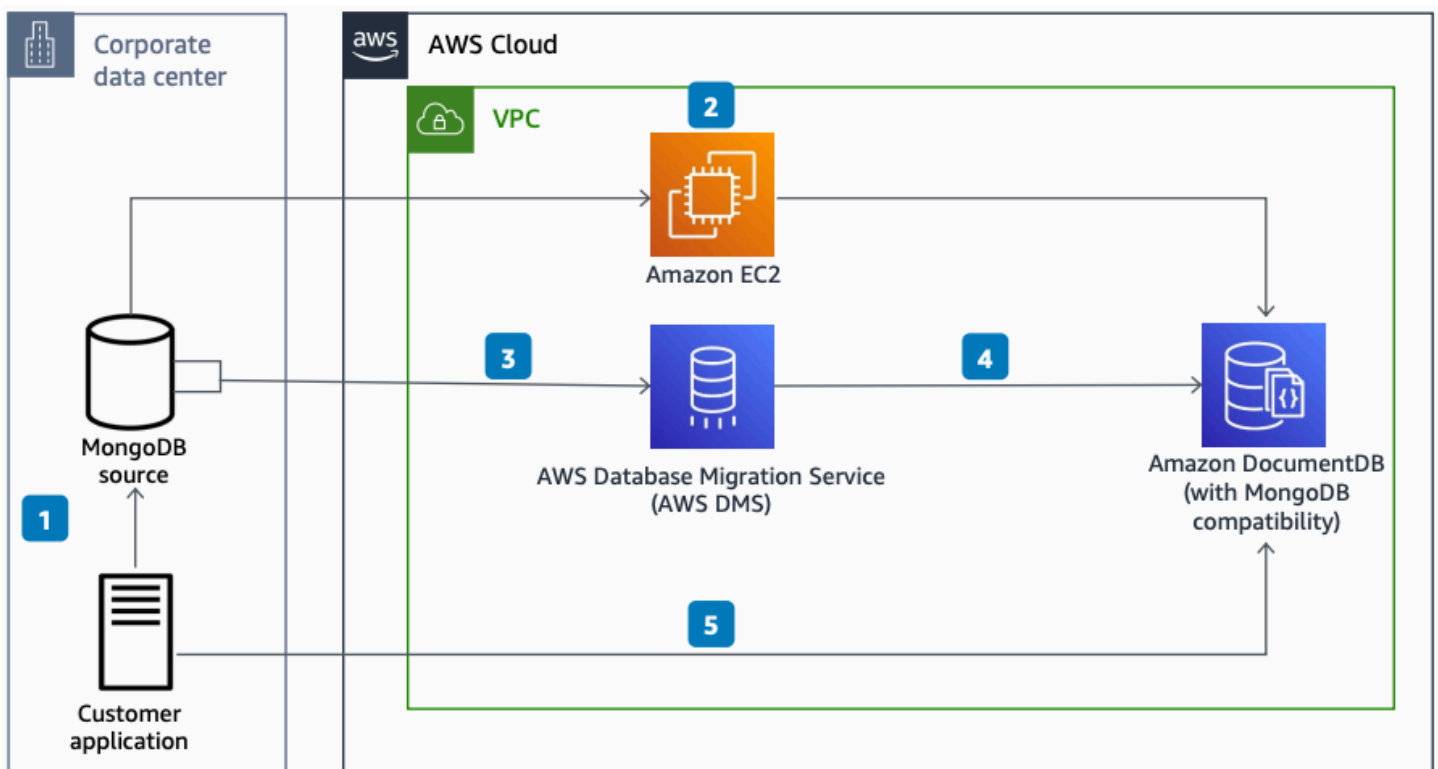
Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.

Migration from MongoDB to Amazon DocumentDB: Online

This architecture shows how to migrate from MongoDB to Amazon DocumentDB using the online approach with [AWS Database Migration Service](#) (AWS DMS). This approach delivers near-zero downtime with medium complexity.

Migration from MongoDB to Amazon DocumentDB: Online Approach



The following steps describe the architecture:

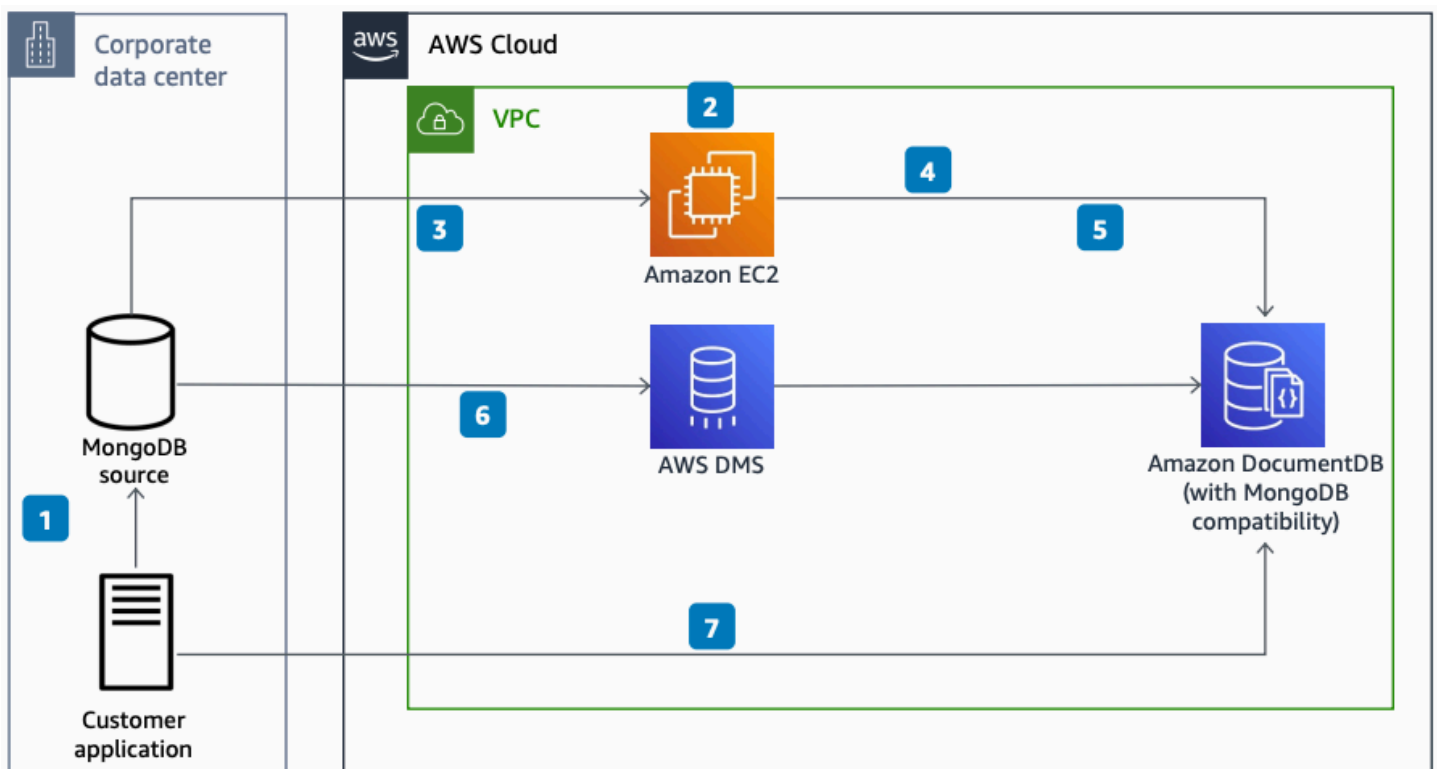
1. When you use the online method to migrate to Amazon DocumentDB, your application continues to write to the source MongoDB database.
2. Create an Amazon EC2 instance in the same VPC as your Amazon DocumentDB cluster and install the mongo shell. Establish connectivity from the Amazon EC2 instance to your MongoDB source, then export the data indexes to the Amazon EC2 migration instance using the mongodump tool.

3. AWS DMS enables database migration using full data load and change data capture (CDC). The online migration approach uses AWS DMS to perform a full data copy and uses CDC to replicate delta changes to Amazon DocumentDB. To improve the performance of AWS DMS full data load migrations, use the auto segmentation or range segmentation option.
4. Use the Amazon EC2 instance created in step 2 to connect to the Amazon DocumentDB cluster to migrate indexes. Use the Amazon DocumentDB Index Tool to check the dumped indexes for compatibility, and pre-create the indexes on the target Amazon DocumentDB cluster.
5. When the data is in sync, switch your application's database connection string to use the new target Amazon DocumentDB cluster.

Migration from MongoDB to Amazon DocumentDB: Hybrid

This architecture shows how to migrate from MongoDB to Amazon DocumentDB using the hybrid approach with MongoDB utilities and AWS DMS. This approach delivers near-zero downtime, and it is the suggested option for source datasets greater than one terabyte (TB).

Migration from MongoDB to Amazon DocumentDB: Hybrid Approach



The following steps describe the architecture:

1. In the hybrid method to migrate to Amazon DocumentDB, your application continues to write to the source MongoDB database.
2. Create an Amazon EC2 instance in the same VPC as your Amazon DocumentDB cluster and install the mongo shell. Establish connectivity from the Amazon EC2 instance to your MongoDB source. Export the data indexes to the Amazon EC2 migration instance using the Amazon DocumentDB Index Tool.

3. Export the data from your MongoDB replica set to the Amazon EC2 migration instance using the mongodump tool. To reduce the impact of the mongodump on the source, set the readPreference option to secondary to force the dump to connect to a secondary replica set member.
4. Use the Amazon DocumentDB Index Tool to restore the indexes that you exported (step 2) in your target Amazon DocumentDB cluster.
5. To restore the data that you dumped in your target cluster in step 3, use the mongorestore utility.
6. AWS DMS enables database migration using full data load and CDC. The hybrid migration approach uses CDC to replicate changes to Amazon DocumentDB.
7. When the data is in sync, change your application's database connection string to use Amazon DocumentDB cluster.