AWS Whitepaper

Getting Started with Amazon DocumentDB (with MongoDB Compatibility)

Getting Started with Amazon DocumentDB (with MongoDB Compatibility): AWS Whitepaper

Table of Contents

Getting Started with Amazon DocumentDB (with MongoDB Compatibility)

Publication date: **June 23, 2021** (*Document revisions*)

Abstract

Amazon DocumentDB (with MongoDB compatibility) is a fast, scalable, highly available, and fully managed document database service that supports MongoDB workloads. This whitepaper covers the architecture and key features of Amazon DocumentDB, and helps you understand how you can use Amazon DocumentDB to run large, mission-critical MongoDB workloads. This whitepaper also covers Amazon DocumentDB security, scalability, performance, and approaches to migrate to Amazon DocumentDB..

The target audience of this whitepaper is solutions architects, database administrators, and developers with a basic understanding of cloud computing, Amazon Web Services (AWS), and MongoDB.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

When developing modern applications, document databases such as MongoDB are a popular choice for storing semi-structured data for use cases like product catalogs, user profiles, mobile applications, and content management. These databases can grow to multiple terabytes in size and may need to scale to millions of reads per second. Setting up and managing large, highly available, high-performance MongoDB databases on your own can be complex and challenging.

Amazon DocumentDB is a fully managed database service that is MongoDB compatible. Amazon DocumentDB is designed to give you the performance, scalability, and availability you need when operating mission-critical MongoDB workloads. You can use the same MongoDB application code, drivers, and tools you do today to run and manage your workloads on Amazon DocumentDB.

Key features of Amazon DocumentDB

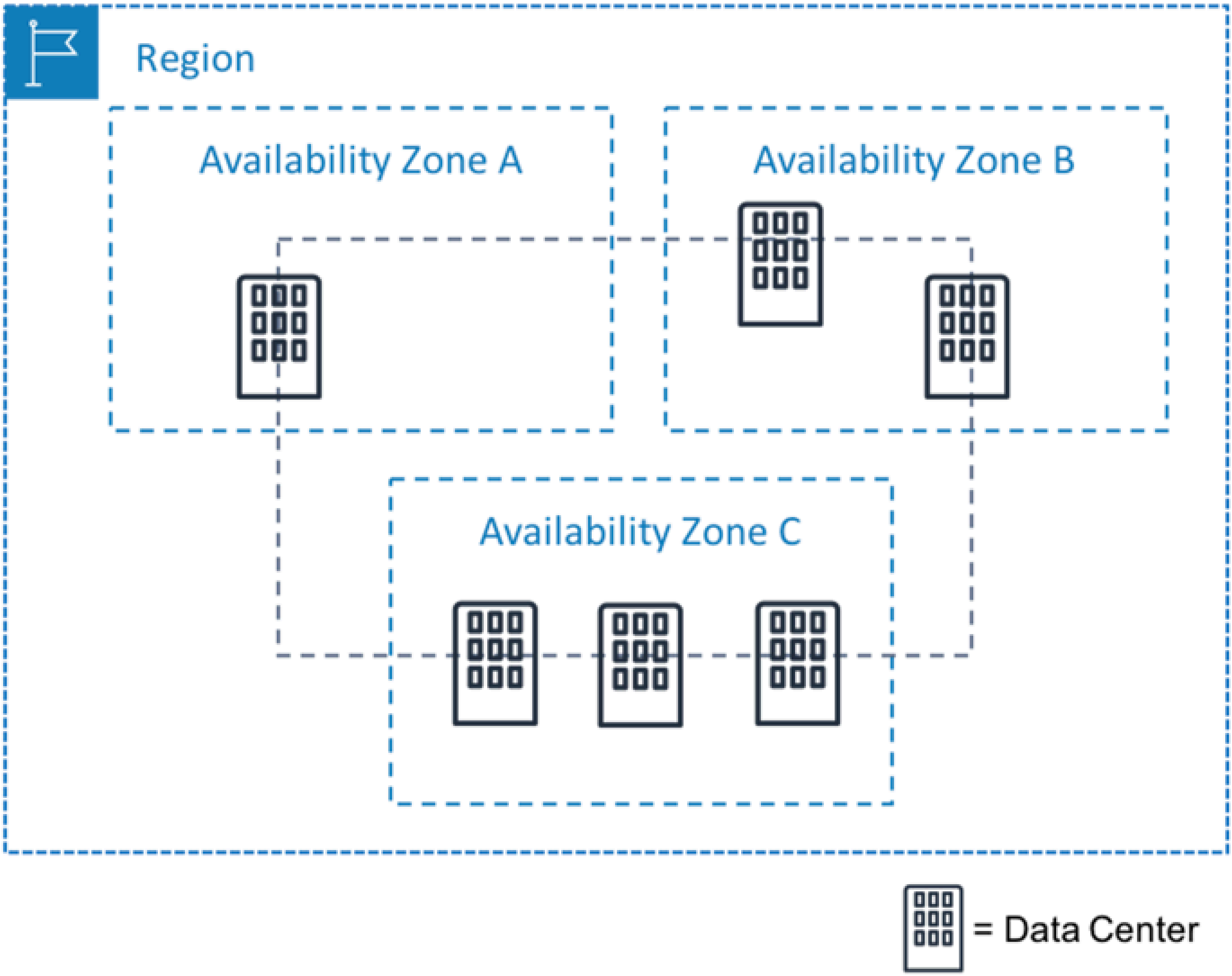
- **Fully managed service** — AWS takes care of the hardware provisioning, patching, backups, high availability, and durability. This frees you from time-consuming administration tasks and lets you focus on building your applications.
- **Compatible with MongoDB** — Amazon DocumentDB is MongoDB-compatible and implements the MongoDB 3.6 and 4.0 API. It emulates the responses a MongoDB client expects from a MongoDB server. This means that you can continue to use your existing MongoDB drivers and tools with Amazon DocumentDB with little or no change. Updating your application to use Amazon DocumentDB could be as simple as redirecting the application to the Amazon DocumentDB endpoint after migrating your data.
- **Highly scalable** — In Amazon DocumentDB, storage and compute are decoupled, and can be scaled independently. You can start with a cluster containing one instance, and add up to 15 read replicas to support millions of reads per second. You do not have to provision storage in advance —Amazon DocumentDB automatically scales provisioned storage in 10 GB segments up to 64 TB as your data grows.
- **Fault tolerant** — Amazon DocumentDB is highly durable. Your data is replicated six ways across three Availability Zones. Amazon DocumentDB transparently handles the loss of up to two out of six data copies without losing write availability, or three out of six copies without losing read availability.
- **High performance** — Amazon DocumentDB uses an all SSD, log-structured storage engine that is purpose- built for database workloads. Amazon DocumentDB delivers twice the throughput of currently available managed MongoDB services.
- **Automatic, continuous, incremental backups and point-in-time recovery** — The Amazon DocumentDB backup capability enables point-in-time recovery for your clusters. You can restore your data to a new cluster to any specified second within your backup retention period, up until the last five minutes. Your automatic backup retention period can be configured up to thirty-five days. Automated backups are stored in Amazon Simple Storage Service (Amazon S3), which is designed for 99.999999999% durability. Amazon DocumentDB backups are automatic, incremental, and continuous, and have no impact on cluster performance.
- **Near instant crash recovery** — Amazon DocumentDB uses log-structured storage and does not require crash recovery replay of database redo logs, greatly reducing restart times (typically less than 30 seconds).

- **Highly secure** — Amazon DocumentDB runs in your Amazon Virtual Private Cloud (Amazon VPC), and encrypts connections using Transport Layer Security (TLS) to secure data in transit. Amazon DocumentDB also enables encryption of data at rest in the default configuration.

AWS Regions and Availability Zones

To understand the architecture of Amazon DocumentDB and how it helps you design highly available applications on AWS, you must be familiar with AWS global infrastructure.

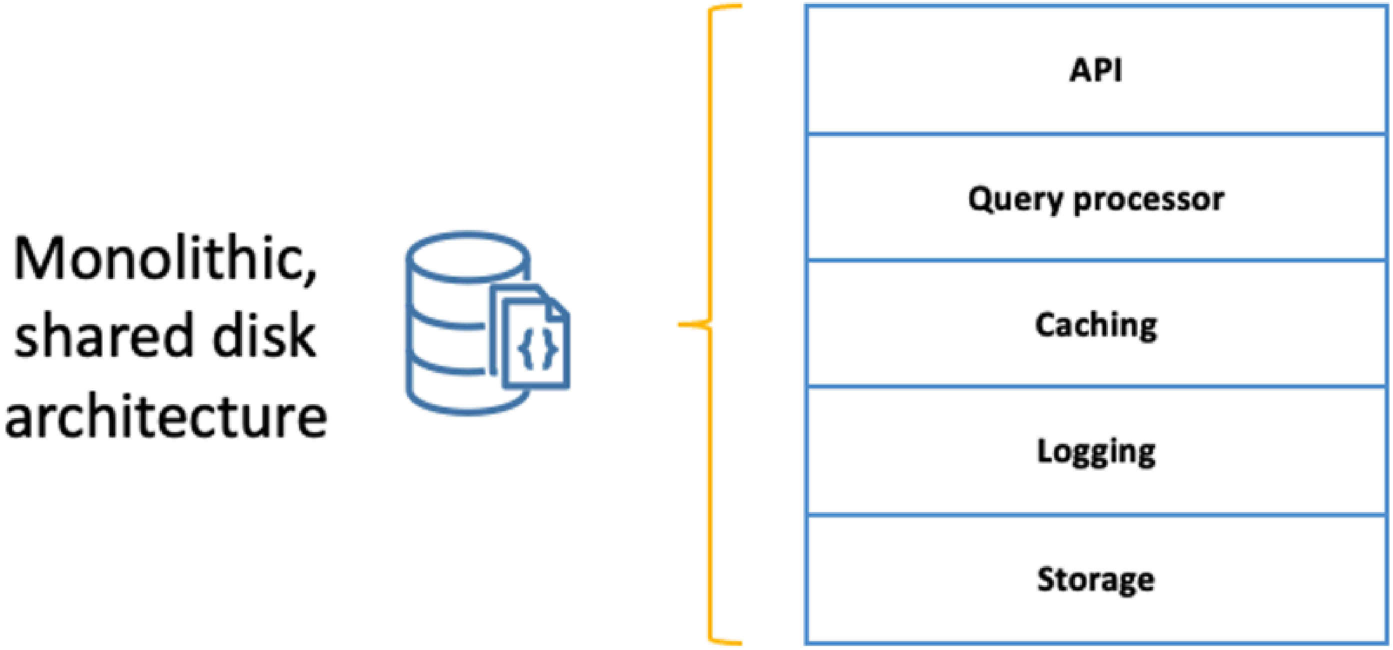
The AWS Global Infrastructure comprises AWS Regions and Availability Zones. AWS Regions are separate geographic areas. *AWS Regions* consist of multiple, physically separated and isolated Availability Zones that are connected with low latency, high throughput, highly redundant networking. *Availability Zones* consist of one or more discrete data centers, each with redundant power, networking, and connectivity, and housed in separate facilities.



AWS Regions and Availability Zones

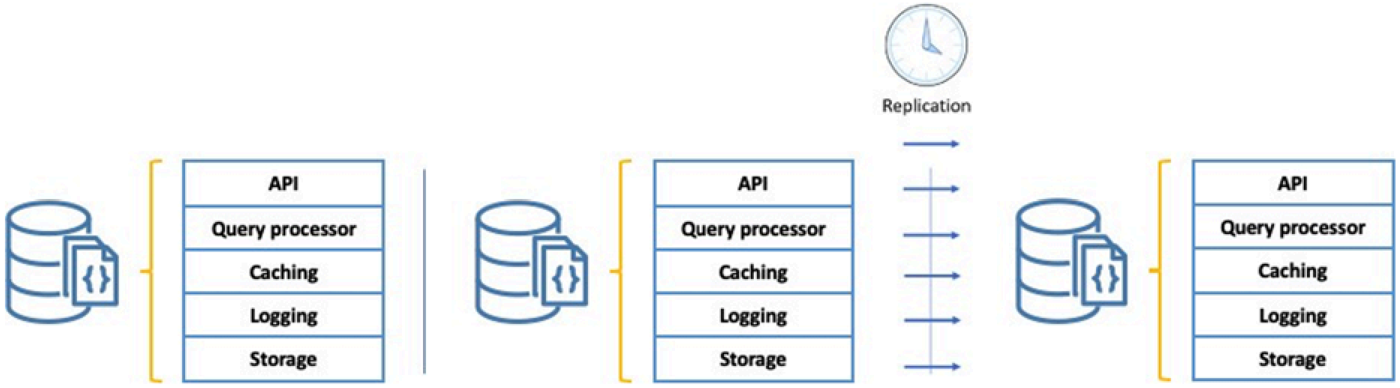
These Availability Zones enable you to operate production applications and databases that are more highly available, fault tolerant, and scalable than possible when using a single data center. You can deploy your applications and databases across multiple Availability Zones. In the unlikely event of a failure of one Availability Zone, user requests are routed to your application instances in the second Availability Zone. This approach ensures that your application continues to remain available at all times.

Limitations of traditional architectures



Architecture of traditional databases

Traditional databases have monolithic architectures—the compute and storage layers are tightly coupled and cannot be scaled independently. Scalability is handled by adding more nodes, each with its own compute and storage. Adding an extra node for scaling or replacing a failed node requires that you copy or replicate the existing data to the new node; this process can take hours, days, or even weeks for large databases.



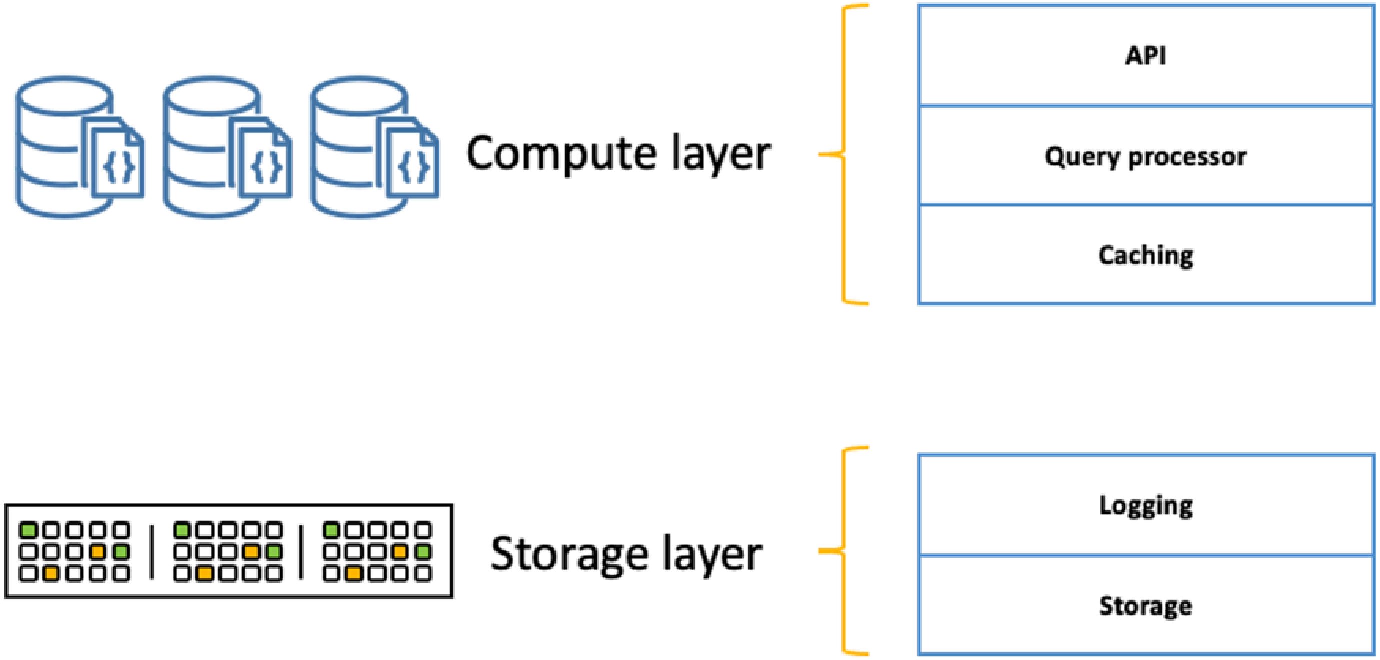
Copying data to a newly added node on traditional database (time consuming)

Amazon DocumentDB: Cloud-native architecture

Amazon DocumentDB is designed for the cloud and to avoid the limitations of traditional database architectures.

Decoupled compute and storage

The compute and storage layers are decoupled in Amazon DocumentDB, and can be scaled independently. The primary instance and replicas share the same cluster volume. Adding a read replica or replacing a failed instance does not require copying any data, and can be performed in a few minutes regardless of the size of your data.

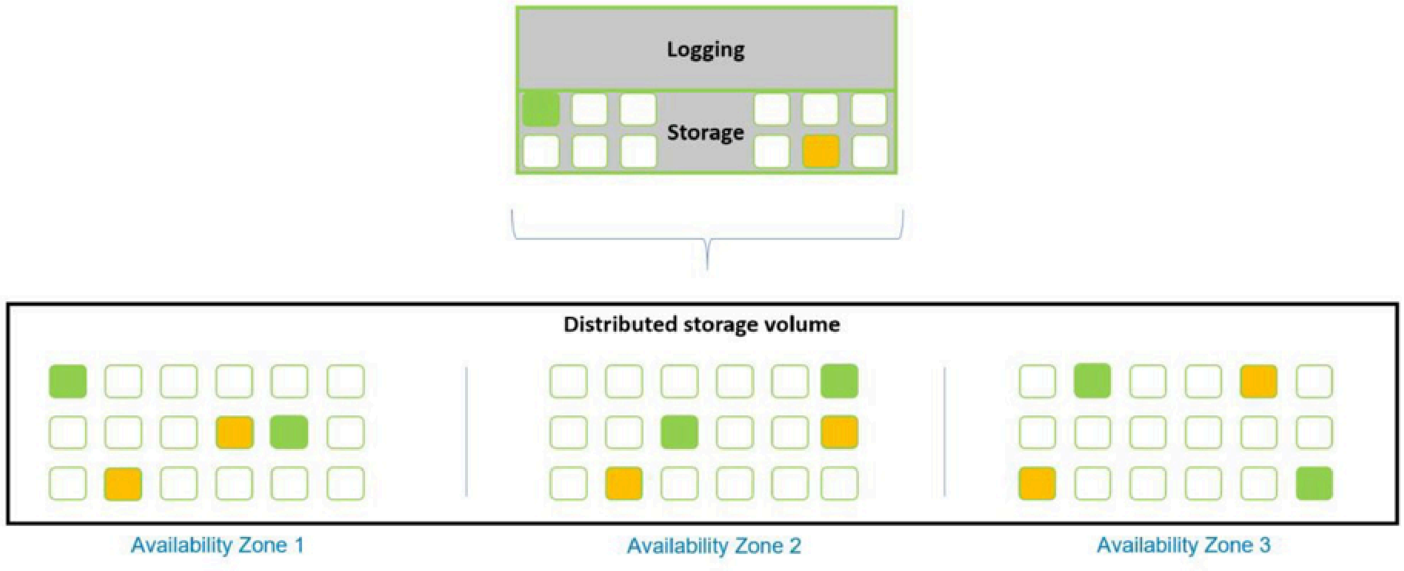


Amazon DocumentDB: Decoupled compute and storage

Fault-tolerant design

In Amazon DocumentDB, the durability is handled at the storage layer. Whether your cluster contains a single instance or 16 instances, you have the same level of durability for your data.

Amazon DocumentDB divides its storage volume into 10-GB segments, each distributed across the cluster, thus isolating the blast radius of disk failures. Each segment is replicated six ways across three Availability Zones.

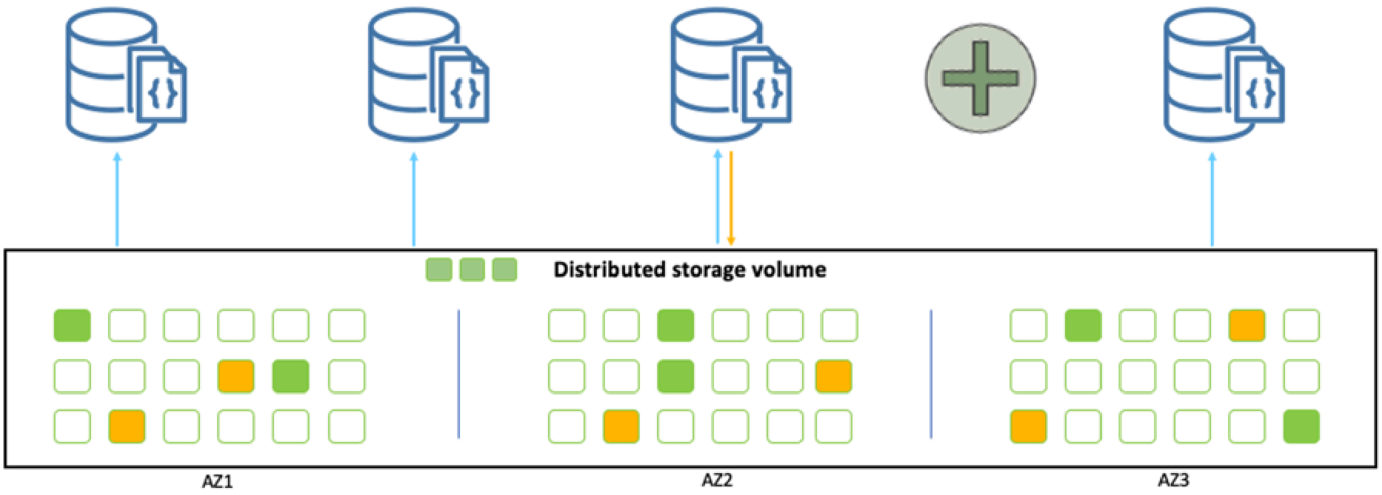


Data replicated six ways across three Availability Zones

Amazon DocumentDB storage is also self-healing; data blocks and disks are continuously scanned for errors and replaced automatically. Amazon DocumentDB monitors disks and storage nodes for failures and automatically replaces or repairs the disks and storage nodes without the need to interrupt read or write processing from the database.

Low-latency read replicas

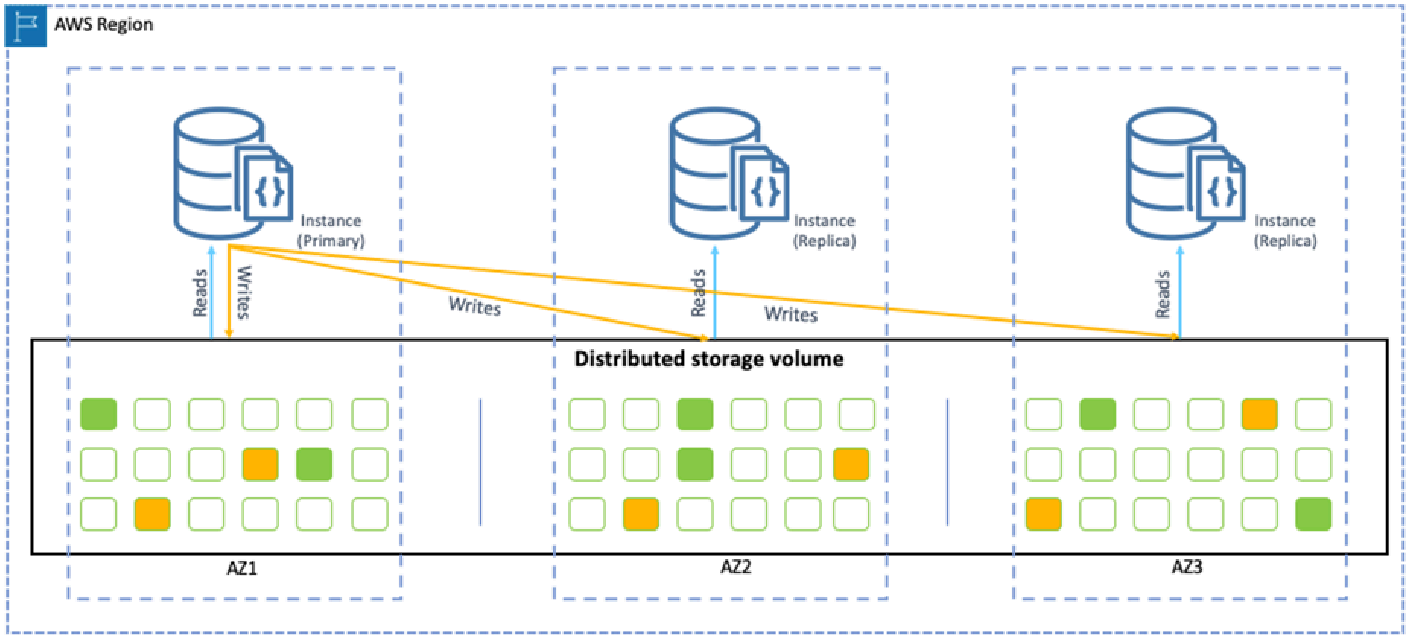
You can create up to 15 Amazon DocumentDB replicas across multiple Availability Zones to scale your read traffic. Amazon DocumentDB replicas share the same underlying storage as the source instance, avoiding the need to copy data to replicas to keep them in sync. This approach frees up more processing power to serve read requests and reduces the replica lag time—typically under 100 milliseconds. As the primary instance and replicas share the same storage, adding a replica does not require any data to be copied. You can add a replica within minutes regardless of the size of your data.



Add up to 15 replicas in minutes regardless of size of data

Amazon DocumentDB architecture

The following diagram shows the main components of Amazon DocumentDB.



Amazon DocumentDB architecture

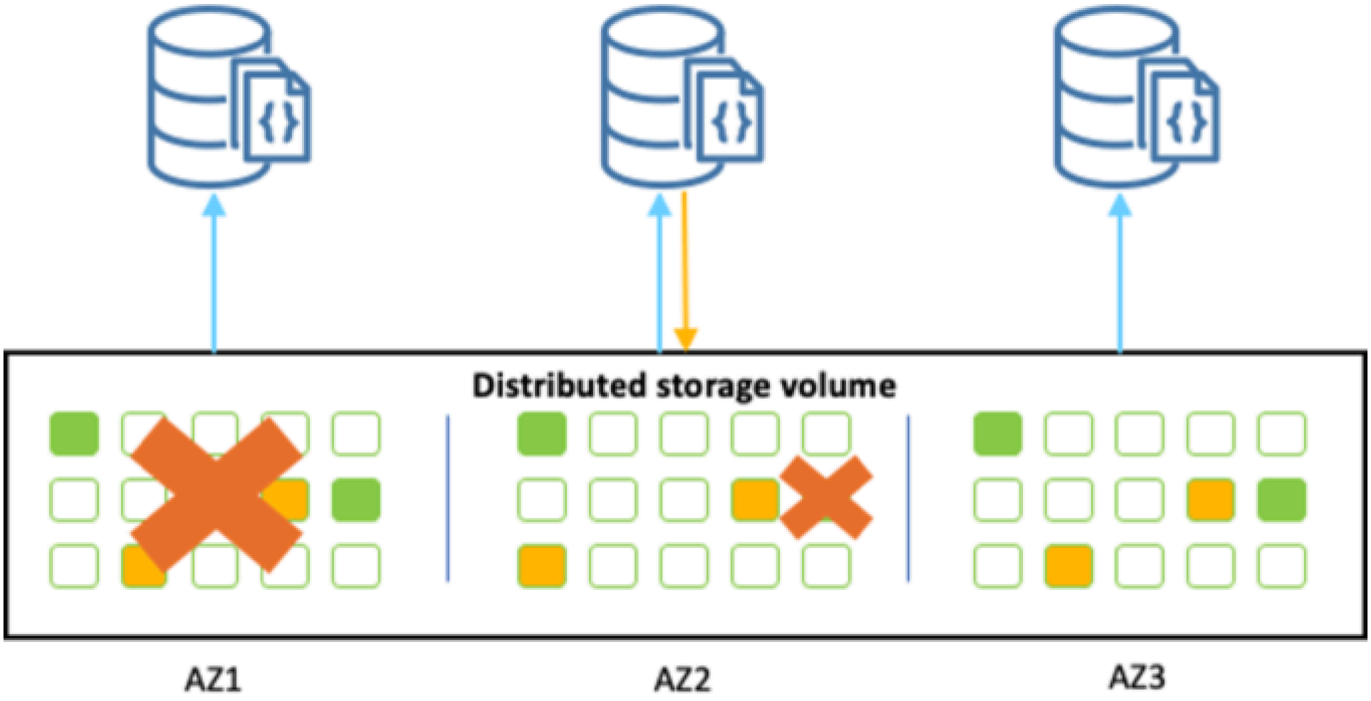
- **Cluster** — A cluster consists of one or more instances that provide the compute, and a cluster volume that manages the data for the instances. A cluster can have up to 16 instances (a primary and up to 15 read replicas). Cluster instances need not be all of the same instance size.
- **Primary instance** — An instance that supports read/write workloads and performs all the data modifications to the cluster volume. Each Amazon DocumentDB cluster has one primary instance.
- **Cluster volume** — The cluster volume provides SSD-backed distributed storage for your database. The primary instance and any Amazon DocumentDB replicas share the same cluster volume.
- **Replicas** — An Amazon DocumentDB replica supports only read operations, and each DB cluster can have up to 15 Amazon DocumentDB replicas. In case the primary instance fails, one of the Amazon DocumentDB replicas is promoted as the primary.

High availability

Amazon DocumentDB has a number of features that make it highly available.

Highly available distributed storage

Amazon DocumentDB replicates your data six ways across three Availability Zones. The cluster volume spans three Availability Zones in a single AWS Region, and each Availability Zone contains two copies of the cluster volume data. This functionality means that Amazon DocumentDB can transparently handle the loss of up to two data copies or an Availability Zone failure without losing write availability, or the loss of up to three data copies without losing read availability.



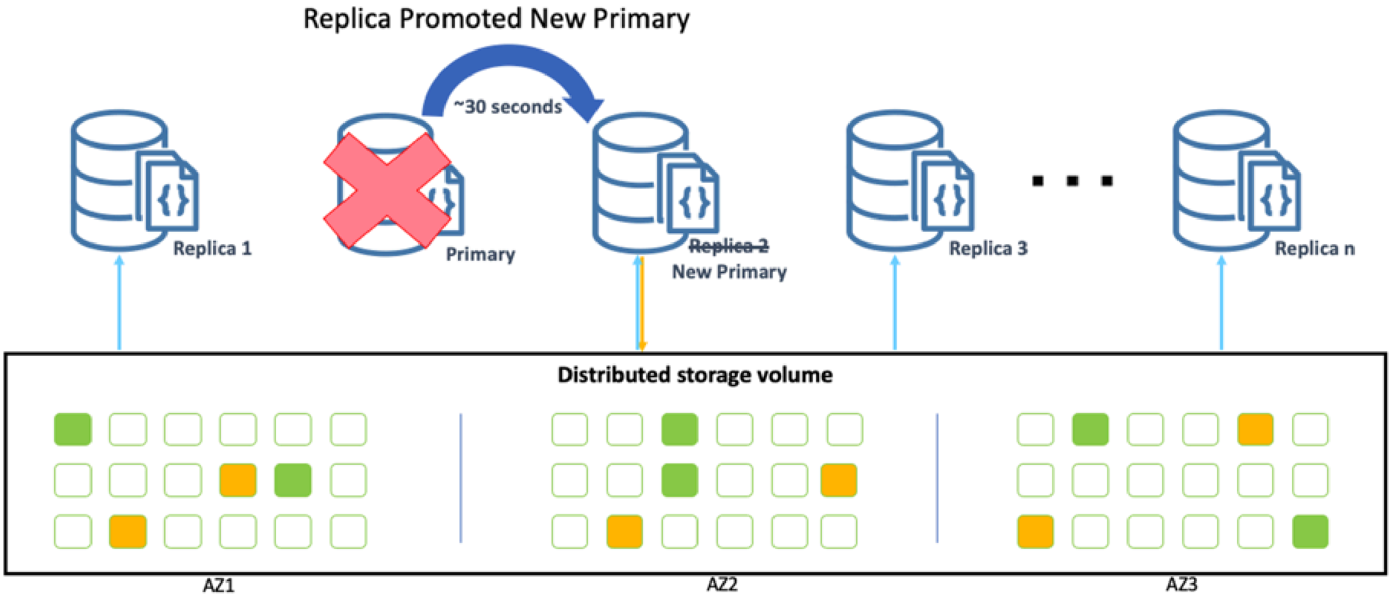
Amazon DocumentDB tolerates loss of three copies of data

Near instant crash recovery

Amazon DocumentDB is designed to recover from a crash almost instantaneously. Unlike other databases, Amazon DocumentDB does not need to replay redo logs after a crash before making the database available for operations. The storage is organized in many small segments and Amazon DocumentDB can perform crash recovery asynchronously on parallel threads. This approach reduces database restart times to less than 30 seconds in most cases.

Automatic failover without data loss

Amazon DocumentDB uses automated checks to detect failure of the primary instance in a cluster. If the primary instance fails, Amazon DocumentDB automatically fails over to any of up to 15 Amazon DocumentDB replicas with minimal downtime and availability impact on applications (up to 30 seconds of downtime for your cluster, but often less time than that). For higher availability, when you create a cluster using the Amazon DocumentDB console, and you choose to create a replica in a different Availability Zone, Amazon DocumentDB creates two instances. It creates the primary instance in one Availability Zone and the replica instance in a different Availability Zone. Failover happens with no data loss, and redo log replay is not required, because the replicas and the primary instance share the same storage.



Any of 15 replicas can be promoted as the primary without data loss

The following table gives guidelines on configurations for meeting different availability goals for your Amazon DocumentDB database.

Table 1 — Typical deployment configurations

Availability target	Total instances	Replicas	Availability Zones	Recovery time
99%	1	0	1	8-10 minutes

Availability target	Total instances	Replicas	Availability Zones	Recovery time
99.9%	2	1	2	<30 seconds
99.99%	3	2	3	<30 seconds
99.99%	4	3	3	<30 seconds

Failover tiers

Each Amazon DocumentDB replica instance is associated with a failover tier (0–15). When a failover occurs due to maintenance or an unlikely hardware failure, the primary instance fails over to a replica with the lowest numbered priority tier. More than one Amazon DocumentDB replica can share the same priority, resulting in promotion tiers. If two or more Amazon DocumentDB replicas share the same priority, then the replica that is largest in size is promoted to primary.

If two or more Amazon DocumentDB replicas share the same priority and size, an arbitrary replica in the same promotion tier is promoted. By setting the failover tier for a group of select replicas to 0 (the highest priority), you can ensure that a failover promotes one of the replicas in that group. Further, you can effectively prevent specific replicas from being promoted to primary if there is a failover by assigning a low-priority tier (high number) to these replicas. This is useful in cases where specific replicas are serving high reads to an application and failing over to one of them would negatively affect a critical application.

High performance

Amazon DocumentDB scales to millions of requests per second with millisecond latencies, and achieves twice the throughput of currently available MongoDB managed services. It uses a number of optimizations to achieve this.

Log structured storage

The database engine is tightly integrated with an SSD-based, virtualized storage layer purpose-built for database workloads, reducing write operations to the storage system.

Tasks related to replication, log processing, and backups are offloaded to the storage layer, reducing the load on compute instances.

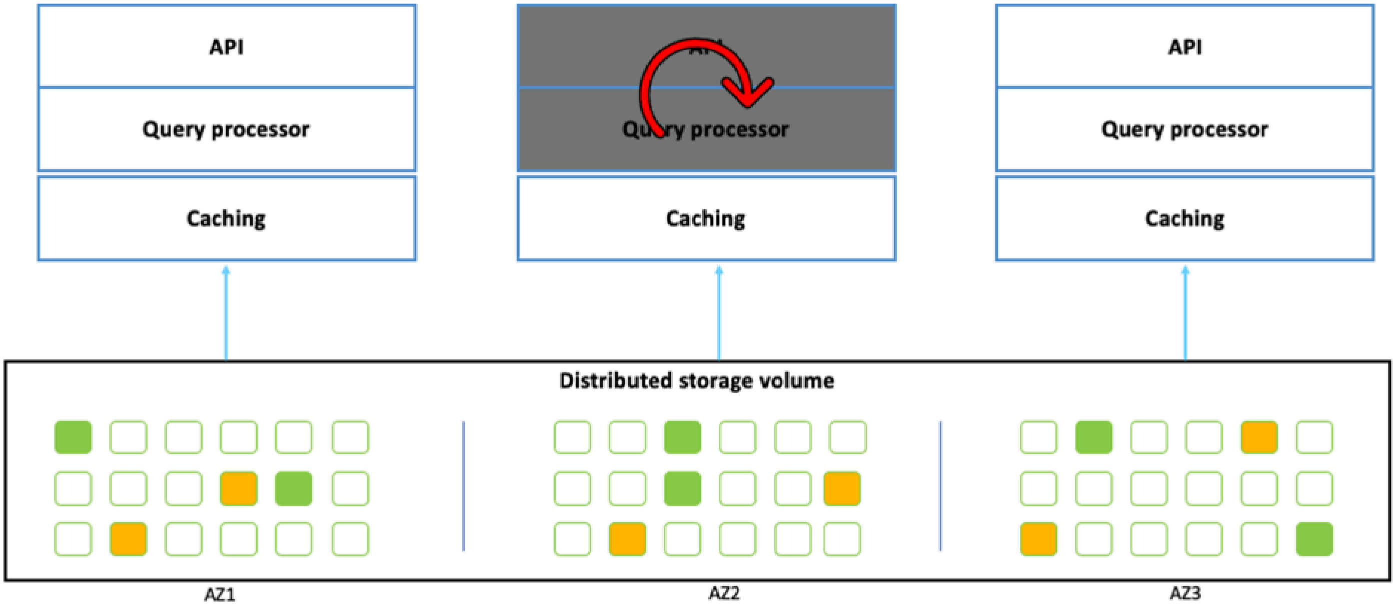
Unlike traditional databases, where the compute node must periodically checkpoint data and flush dirty blocks from buffers to disk, in Amazon DocumentDB only the write-ahead log records are written to storage. This reduces unnecessary communication between the compute and storage, enabling more efficient use of network I/O.

Quorum-based reads and writes

I/O operations use distributed systems techniques such as quorums to improve performance consistency and tolerance to outliers. By default, data write operations are acknowledged as soon as they are committed by four out of six storage nodes, and individual storage nodes acknowledge the write operations as soon as the log records are persisted to disk. This behavior cannot be changed. A slow or failed storage node does not impact database performance or availability due to the use of the quorum model.

Survivable caches

In Amazon DocumentDB, the database buffer cache has been moved out of the database process. If a database restarts, the cache remains warm, and performance is not impacted due to a cold cache, as is the case with traditional databases. This approach lets you resume fully loaded operations much faster.

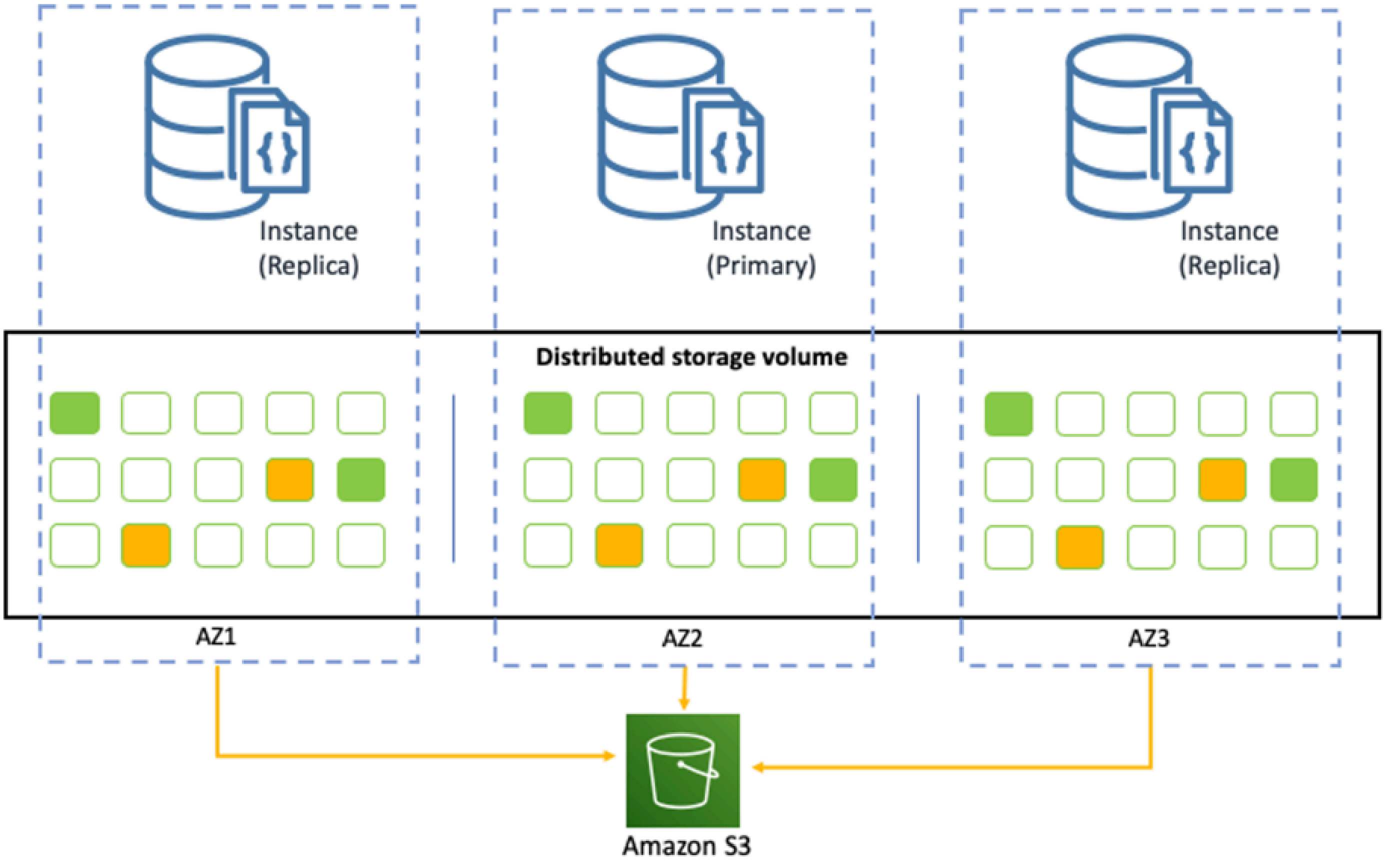


Cache is separate from database and survives database restart

Additionally, the buffer cache is managed independently on each instance. If a replica instance does not have the data resident in memory, the buffer cache is not polluted allowing each replica to manage their own buffer cache.

Automatic, nearly continuous backups

Amazon DocumentDB continuously backs up data to Amazon S3, which is designed for 99.999999999% durability. Amazon DocumentDB backups are automatic, incremental, and continuous, and have no impact on database performance, as the backup is offloaded to the storage layer. By default, backup and the ability to perform a point-in- time restore is enabled on all Amazon DocumentDB clusters.



Backups are offloaded to the storage layer and do not impact performance

The Amazon DocumentDB backup capability enables point-in-time recovery for your instance. This functionality allows you to restore your database to any second during your retention period (up to the last five minutes) with only a few clicks.

Scalability

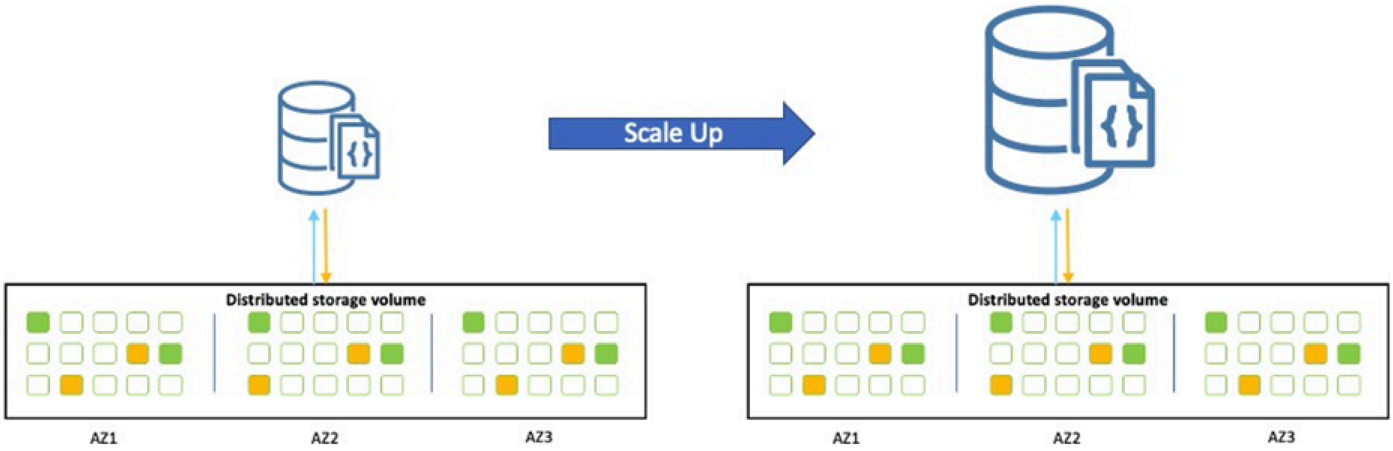
Amazon DocumentDB is designed to be highly scalable. Amazon DocumentDB supports both vertical and horizontal scaling. You can scale vertically by increasing the size of your instances. You can scale horizontally by adding up to 15 read replicas, supporting millions of requests per second. The primary instance and read replicas share the same storage, and read replicas can be added in a few minutes with minimal impact on database availability. Amazon DocumentDB can automatically scale your storage up to 64 TB as your data grows and you only pay for the storage that you use.

Scaling up

Amazon DocumentDB instances are available in various sizes, starting from the db.t3.medium instance with 2 vCPUs and 4-GiB RAM, to the `db.r5.24xlarge` instance with 96 vCPUs and 768-GiB RAM. The complete list of Amazon DocumentDB instance types and regional availability can be found on the [Amazon DocumentDB pricing page].

You choose an appropriate instance type based on the RAM, vCPU, and network throughput required. You can start with a smaller instance type such as `db.t3.medium` or `db.r5.large`, and scale up to a larger instance type as your application grows.

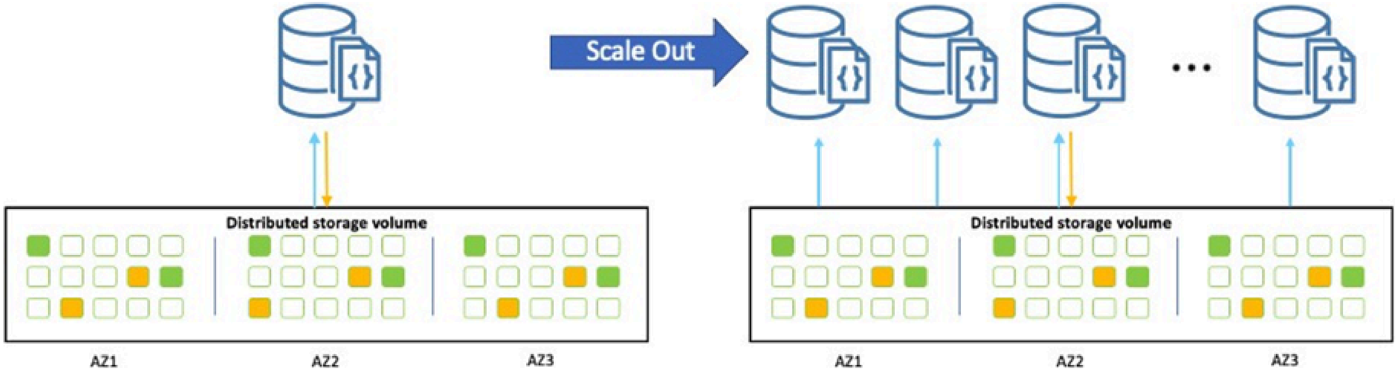
Compute scaling operations typically complete in a few minutes irrespective of the size of your data. Scaling does not require any copying of data because the storage and compute layers are decoupled in Amazon DocumentDB. Scaling up is useful if you want to scale your write capacity or to provision a larger read replica instance for running read-only analytics workloads.



Scale up or scale down in minutes without moving any data

Scaling out

You can scale out your cluster by adding read replicas. You can add up to 15 read replicas and scale your read capacity to millions of requests per second. The replica lag is low (usually less than 100 milliseconds) because the read replicas and the primary instance share the same storage volume. You can add replicas in minutes without any downtime or impact to database performance.



Add up to 15 replicas in minutes without downtime

Automatic scaling storage

With Amazon DocumentDB, unlike traditional databases, you do not have to provision storage space explicitly while creating the database. Amazon DocumentDB data is stored in an SSD-backed virtual volume (cluster volume) that automatically grows as the amount of data in the database increases. Every database page read operation counts as one I/O. Amazon DocumentDB issues reads against the storage layer to fetch pages not present in the buffer cache. Each page is 8KB in Amazon DocumentDB. The volume grows in increments of ten GB up to a maximum of 64 TB. This process is transparent to your application, without any impact on application availability or performance.

Security and compliance

With Amazon DocumentDB, best practices are the default. Authentication, encryption- at-rest, and encryption-in-transit are enabled by default. You can control access to Amazon DocumentDB management operations, such as creating and modifying clusters, instances, and more, using AWS IAM users, roles, and policies. You can authenticate users to an Amazon DocumentDB database via standard MongoDB tools and drivers.

AWS IAM

Amazon DocumentDB is integrated with AWS Identity and Access Management (IAM) and provides you the ability to control the actions that your AWS IAM users and groups can take on specific Amazon DocumentDB resources, including clusters, instances, and snapshots. In addition, you can enable resource-level permissions by tagging your Amazon DocumentDB resources, and configuring IAM rules based on the tags.

Network security

Amazon DocumentDB clusters are VPC-only and are created directly in your VPC. Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define. Amazon VPC enables you to isolate your cluster in your own virtual network and connect to your on-premises IT infrastructure using industry-standard encrypted IPsec VPNs.

You can also use AWS Direct Connect to create a dedicated, private network connection between your intranet and Amazon VPC. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use multiple layers of security, including security groups and network access control lists (ACLs), to help control access in each subnet. This approach gives you complete control over who can access your Amazon DocumentDB database.

Encryption

Amazon DocumentDB supports TLS to encrypt connections from applications to secure data in transit. Amazon DocumentDB also supports encryption of data at rest using AES-256. Encryption is applied cluster wide and all of the data is encrypted, including the cluster data, indexes, snapshots, logs, and automated backups. For data stored at rest, encryption keys are managed by AWS Key Management Service (AWS KMS), which is a highly available, durable, and secure solution for managing sensitive encryption keys. With AWS KMS, you can use the service-managed key, import existing key material, or create your own encryptions keys.

User management

You can connect to Amazon DocumentDB using standard MongoDB tools and drivers. Amazon DocumentDB supports authentication using the Salted Challenge Response Authentication Mechanism (SCRAM), which is the default authentication mechanism with MongoDB.

When you create an Amazon DocumentDB cluster, you specify a primary user. The primary user has administrative permissions for the cluster. You can connect as the primary user to Amazon DocumentDB and create up to 1,000 users per cluster using `db.createUser`. Additionally, Amazon DocumentDB supports Role-based Access Control (RBAC) that gives you the ability to create users and attach built-in roles to restrict what operations the user has authorization to perform. Common scenarios for using RBAC include enforcing least privilege such as read-only role or building a multi- tenant application where each tenant is restricted to accessing a single database in the cluster.

Auditing events

Amazon DocumentDB supports auditing of the operations performed on your cluster. Once auditing is enabled, Amazon DocumentDB tracks authentication, Data Definition Language (DDL), and user management events. For example, with the auditing feature, you can track failed login attempts, or DDL operations like the creation of collections or indexes. These audit records are exported as JSON documents to Amazon CloudWatch Logs for you to analyze and monitor.

Compliance

Amazon DocumentDB is designed to meet the highest security standards and to make it easy for you to verify our security and meet your own regulatory and compliance obligations. Amazon DocumentDB has been assessed to comply with PCI DSS, ISO 9001, 27001, 27017, and 27018, System and Organization Controls (SOC) 1, 2, and 3, in addition to being HIPAA eligible.

Backup and restore

Amazon DocumentDB backs up your cluster volume automatically and retains backup data for the length of the backup retention period (between one and 35 days). The Amazon DocumentDB backup capability enables point-in-time recovery of your cluster to any second during your retention period, up to the last five minutes, with just a few clicks.

If you want to retain a backup beyond the maximum retention period, you can take a manual snapshot of the cluster. Amazon DocumentDB snapshots are user-initiated full backups of your database that are kept until you explicitly delete them. Additionally, manual cluster snapshots can be shared and copied by authorized AWS accounts. You can share encrypted or unencrypted manual snapshots. When sharing an unencrypted snapshot, authorized AWS accounts can restore the cluster directly from the snapshot instead of making a copy of it and restoring from that. However, you can't restore a cluster from a snapshot that is both shared and encrypted. Instead, you can make a copy of the cluster and restore the cluster from that copy.

Backups are stored in Amazon S3, which is designed for 99.999999999% durability. Backups are automatic, incremental, and continuous. Backups have no impact on database availability or performance because the backups are offloaded to the storage layer.

To restore your data, you can create a new cluster quickly from the backup Amazon DocumentDB maintains, or from a cluster snapshot. Using the automatic backup, you can restore a cluster to any point in time that is within the cluster's backup retention period.

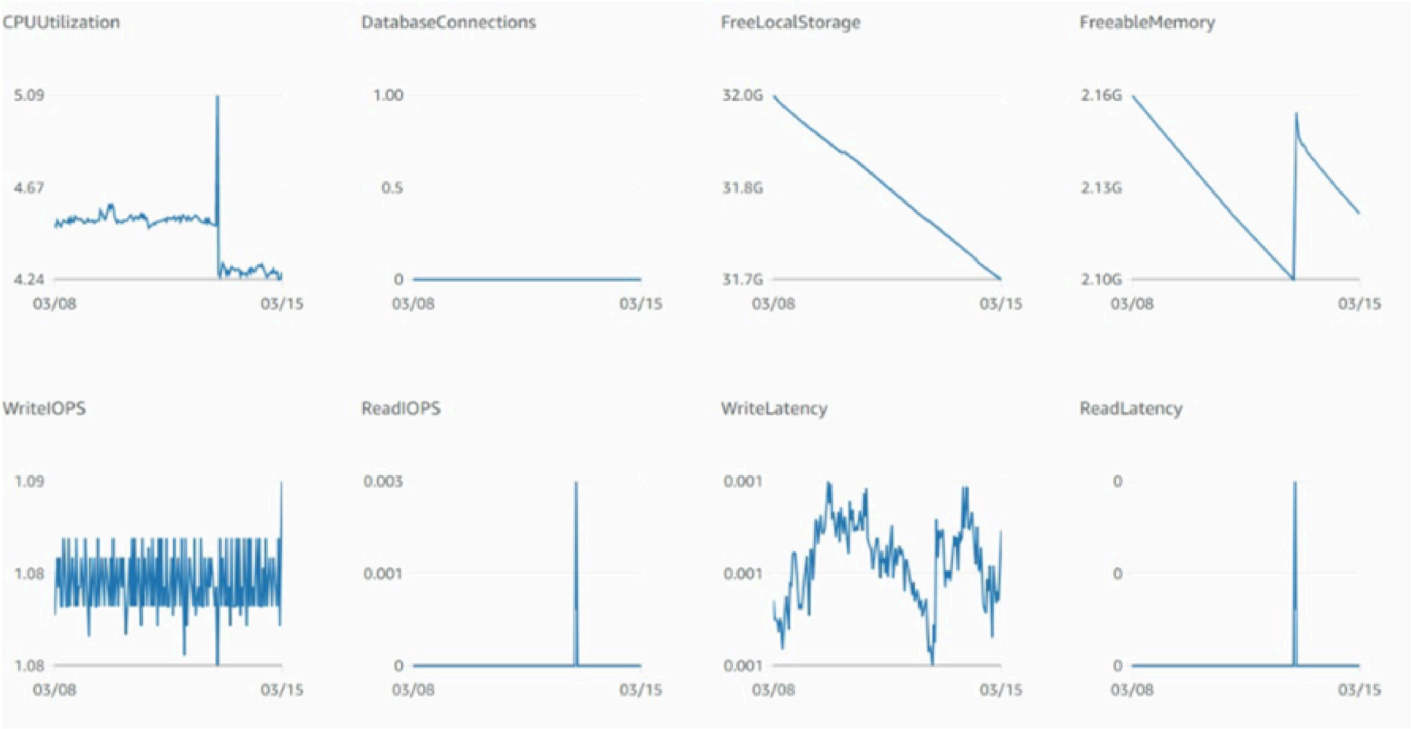
Managing Amazon DocumentDB

You can access and manage your Amazon DocumentDB cluster in several ways. When you are getting started with Amazon DocumentDB, the simplest way is to use the AWS Management Console.

In addition to using the AWS Management Console, you can manage Amazon DocumentDB using the AWS Command Line Interface (AWS CLI), or you can programmatically interact with and manage your Amazon DocumentDB cluster using the AWS SDKs and libraries. AWS SDKs and libraries are available for many popular languages like Python, Node.js, PHP, Go, Java, C#/.NET, R, and Ruby.

Monitoring

You can monitor Amazon DocumentDB using several methods. You can monitor the health and status of your Amazon DocumentDB cluster and your Amazon DocumentDB instances using the AWS Management Console or the AWS CLI. Amazon DocumentDB integrates with Amazon CloudWatch and you can monitor performance metrics like CPU utilization, memory, IOPS, and network throughput using Amazon CloudWatch.



Amazon DocumentDB CloudWatch metrics

Amazon DocumentDB tracks the events related to your cluster. You can view the history of the events including details on snapshot creation, failover, instance reboots, and any modifications to your cluster. You can use the AWS Management Console or the AWS CLI (`describe-events` command) to view these event details.

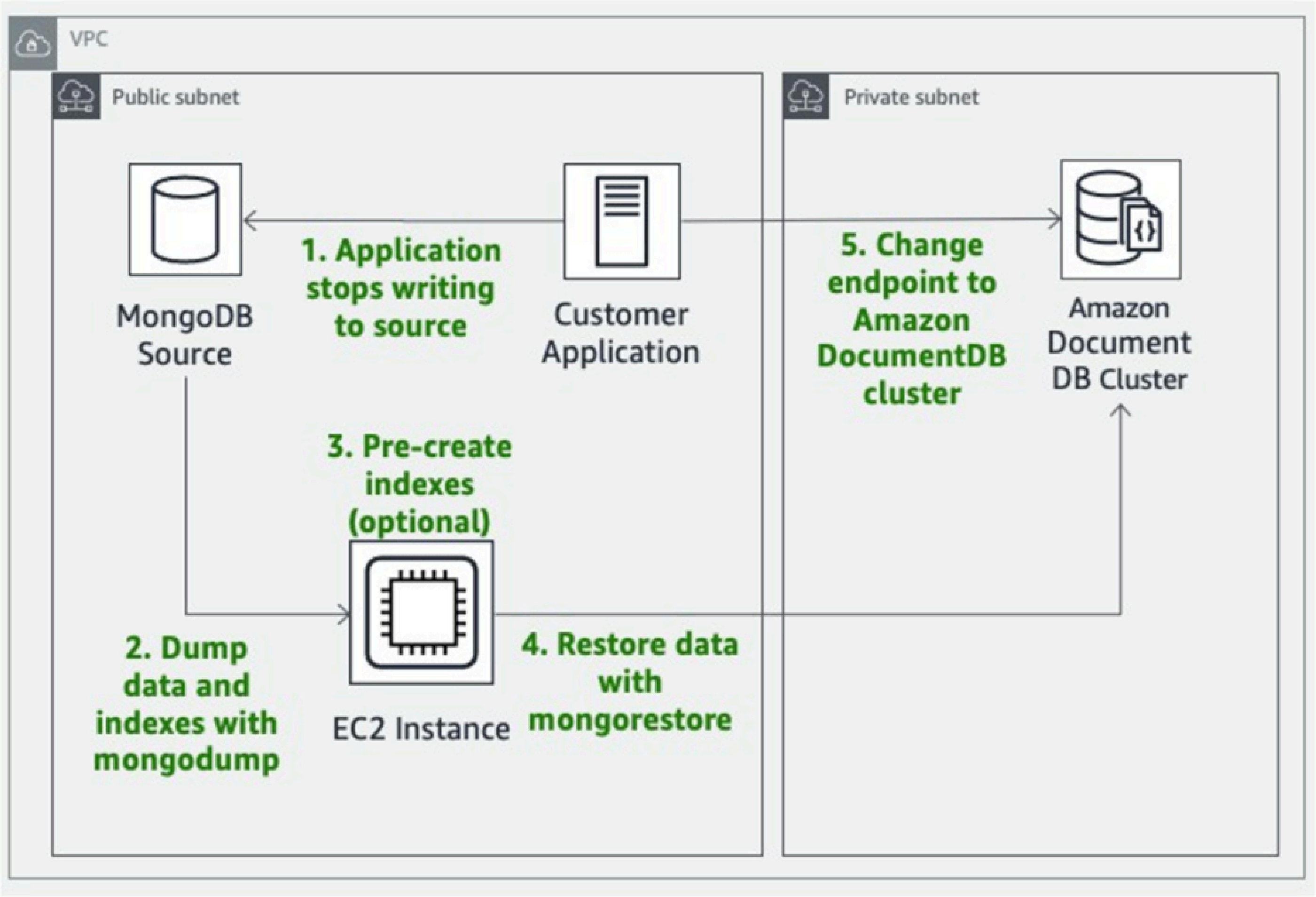
Using Event Subscriptions, you can also sign up to receive notifications for management events that occur on your Amazon DocumentDB clusters, instances, cluster snapshots, and parameter groups. Whenever events occur, Amazon DocumentDB uses Amazon Simple Notification Service (Amazon SNS) to notify the subscriber through supported methods such as an email, a text message, or a call to an HTTP endpoint. With Event Subscriptions you can be notified automatically of events such as the completion of patching, an instance failover, or a parameter group modification. For example, you can use AWS Lambda to process the events of a database instance. Developers can subscribe to over 50 types of events using the AWS Management Console or the CLI.

Migrating to Amazon DocumentDB

You can migrate your data from any MongoDB database, either on-premises or in the cloud (for example, a MongoDB database running on Amazon EC2), to Amazon DocumentDB. You can migrate your data from the source MongoDB database to Amazon DocumentDB using a number of approaches.

Offline migration

The simplest approach is to do an offline migration. Because Amazon DocumentDB is compatible with the MongoDB API, you can use the `mongodump` tool to export the data from MongoDB, and the `mongorestore` tool to restore the data into Amazon DocumentDB. The offline migration method results in downtime while your dump and restore operations are running. This method is suitable for migration of non-production workloads or for migration of non-critical databases where you can afford the downtime.

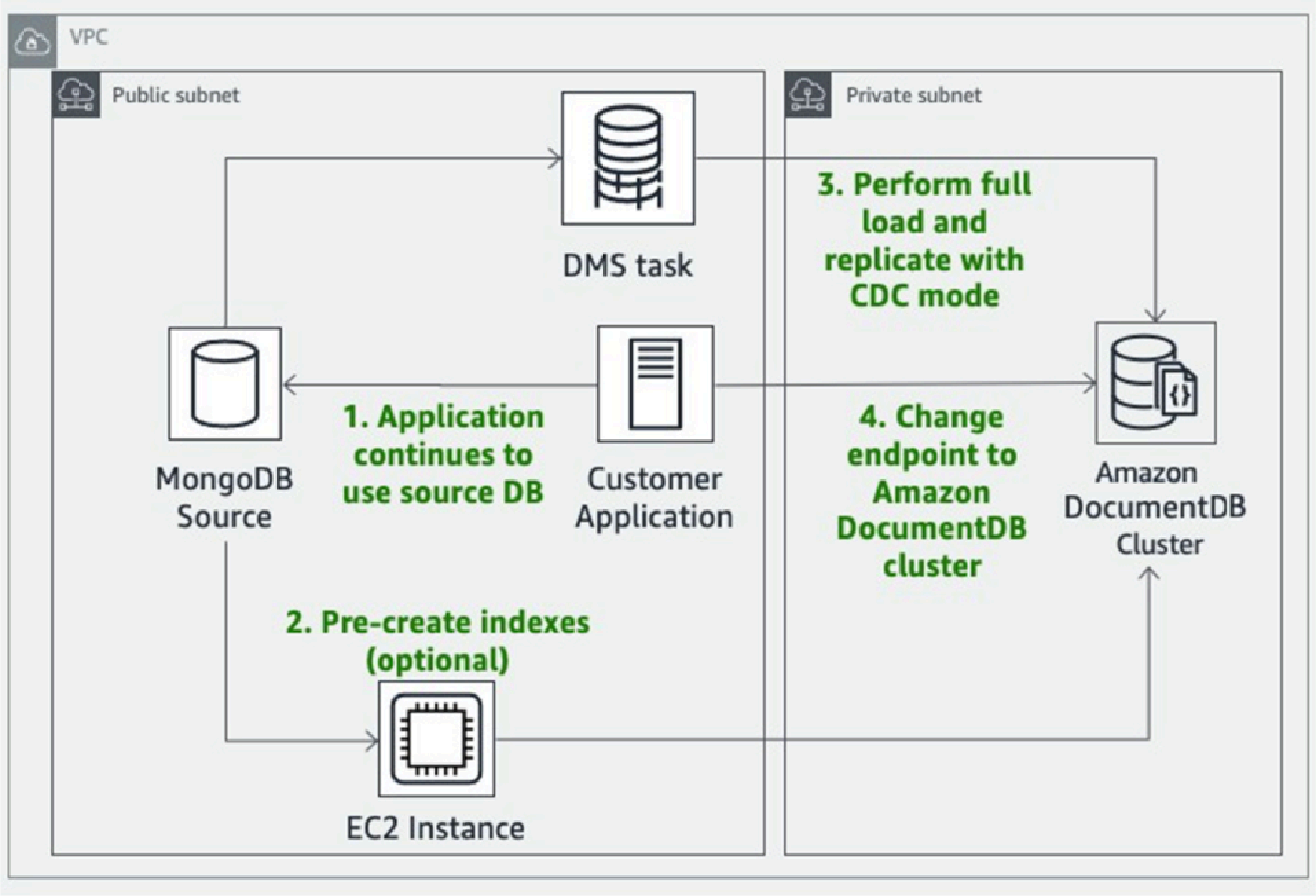

VPC
Public subnet
Private subnet
MongoDB Source
1. Application stops writing to source
Customer Application
5. Change endpoint to Amazon DocumentDB cluster
Amazon Document DB Cluster
3. Pre-create indexes (optional)
2. Dump data and indexes with mongodump
EC2 Instance
4. Restore data with mongorestore

Offline migration approach

Online migration

For migration of production workloads with minimal downtime, you can use the online approach or the hybrid approach. With the online migration approach, you use AWS Database Migration Service (AWS DMS) to migrate the data from MongoDB to Amazon DocumentDB. AWS DMS performs an initial full load of the data from the MongoDB source to Amazon DocumentDB. During the full load, you source database is available for operations. Once the full load is completed, AWS DMS switches to change data capture (CDC) mode to keep the source (MongoDB) and destination (Amazon DocumentDB) in sync. Once the databases are in sync, you can switch your applications to point to Amazon DocumentDB with near zero downtime.

Refer to the AWS Database Migration Service documentation for more information on migrating from MongoDB to Amazon DocumentDB.


VPC
Public subnet
Private subnet
DMS task
3. Perform full load and replicate with CDC mode
MongoDB Source
1. Application continues to use source DB
Customer Application
4. Change endpoint to Amazon DocumentDB cluster
Amazon DocumentDB Cluster
2. Pre-create indexes (optional)
EC2 Instance

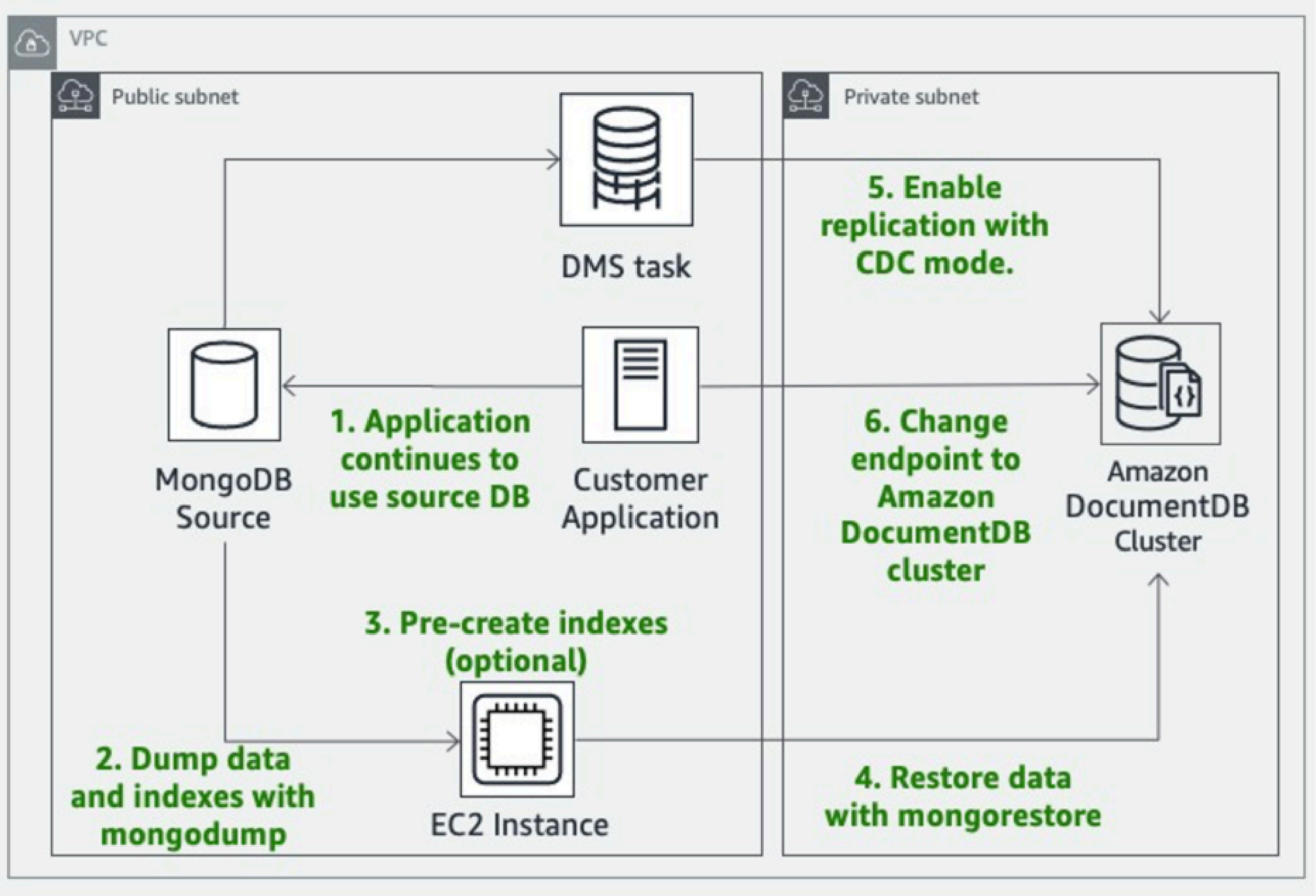
Online migration approach using DMS

Hybrid approach

The hybrid approach is a combination of the offline and online migration approaches. The hybrid approach is useful in a scenario where you need minimal downtime during migration, but the size of the source database is large or sufficient bandwidth is not available to migrate the data in a reasonable amount of time.

The hybrid approach has two phases. In the first phase, you export the data from the source MongoDB using the `mongodump` tool, transfer it to AWS (if the source is on- premises), and restore it to Amazon DocumentDB. You can use [AWS Direct Connect]() or [AWS Snowball]() to transfer the export dump to AWS. During this phase, the source (MongoDB) is available for operations and the data restored to Amazon DocumentDB does not contain the latest changes.

In the second phase, you use AWS DMS in CDC mode to copy the changes from the source (MongoDB) to Amazon DocumentDB and keep them in sync. Once the databases are in sync, you can switch your applications to point to Amazon DocumentDB with near zero downtime.


VPC
Public subnet
Private subnet
DMS task
5. Enable replication with CDC mode.
MongoDB Source
1. Application continues to use source DB
Customer Application
6. Change endpoint to Amazon DocumentDB cluster
Amazon DocumentDB Cluster
3. Pre-create indexes (optional)
2. Dump data and indexes with mongodump
EC2 Instance
4. Restore data with mongorestore

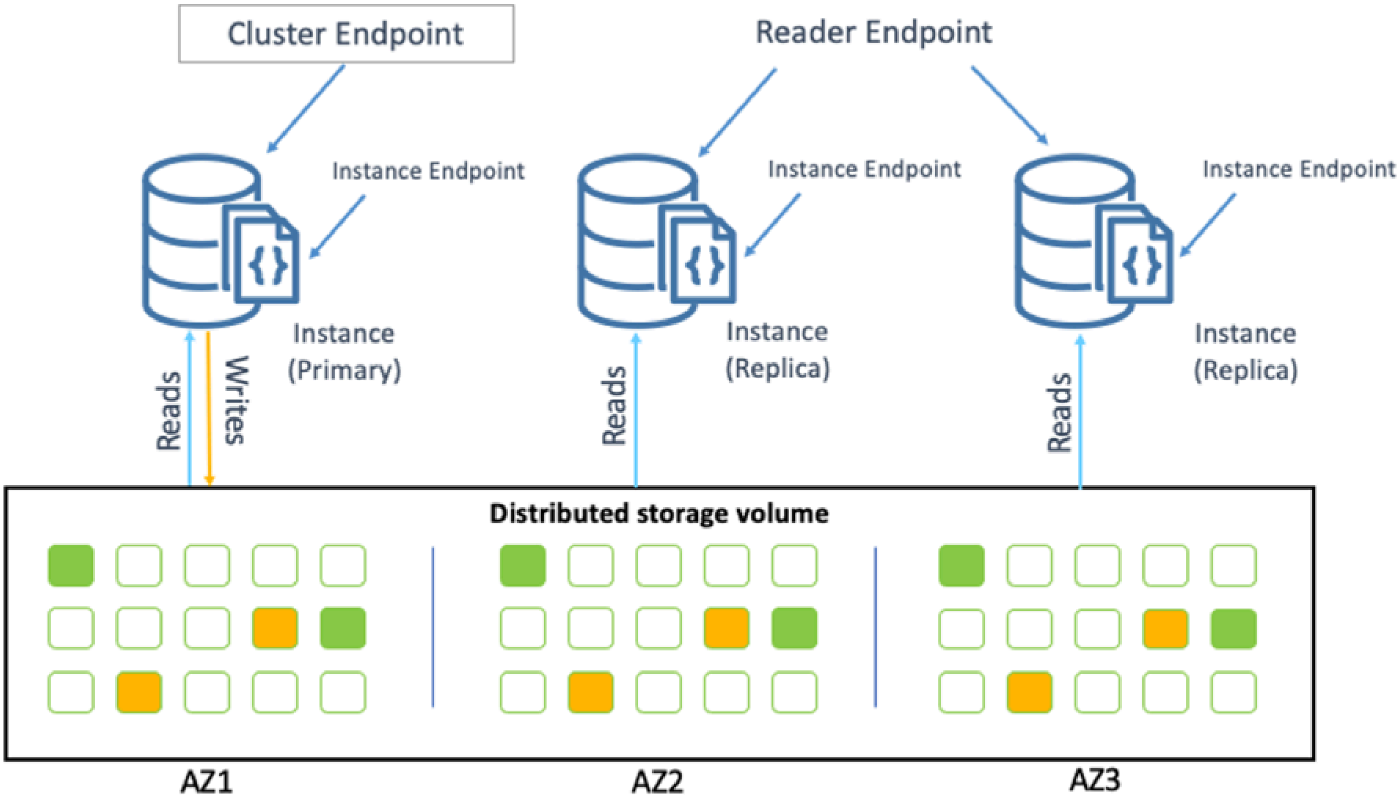
Hybrid migration approach using DMS

Although write operations with existing indexes can be parallelized, foreground and background index builds are single-threaded. Regardless of the approach, pre-creating indexes in your Amazon DocumentDB cluster before importing your data usually results in a faster migration time. AWS DMS also supports parallel full load with the range segmentation option when using Amazon DocumentDB and MongoDB as a source. You can accelerate the migration of large collections by splitting them into segments and loading and unloading the segments in-parallel in the same migration task. This feature could improve the migration performance by up to 3x.

AWS DMS also supports parallel full load with the range segmentation option when using Amazon DocumentDB and MongoDB as a source. You can accelerate the migration of large collections by splitting them into segments and loading and unloading the segments in-parallel in the same migration task. This feature could improve the migration performance by up to 3x.

Connecting to Amazon DocumentDB

Amazon DocumentDB is compatible with the MongoDB 3.6 and 4.0 APIs. With Amazon DocumentDB, you can run the same application code and use the same drivers and tools that you use with MongoDB. For example, you can use the Mongo shell to connect to Amazon DocumentDB and perform operations like creating and editing collections and documents. You connect to Amazon DocumentDB in replica set mode (recommended) or by using the endpoints for your cluster. There are three types of endpoints for Amazon DocumentDB—the cluster endpoint, the reader endpoint, and the instance endpoint.



Connect to Amazon DocumentDB with your existing tools via the endpoints

Replica set mode

When you connect in replica set mode, your Amazon DocumentDB cluster appears to your drivers and clients as a replica set. Connecting to the cluster endpoint in replica set mode is the recommended approach for general use. Replica set mode is advantageous for high availability and effectively balancing client requests in your cluster. To route requests to your replicas, choose a driver read preference setting that maximizes read scaling while meeting your application's read consistency requirements. Instances added and removed from your Amazon DocumentDB cluster are reflected automatically in the replica set configuration. You can connect to your Amazon DocumentDB cluster endpoint in replica set mode by specifying the replica set name `rs0`

Cluster endpoint

The cluster endpoint connects to your cluster's current primary instance. The cluster endpoint can be used for read and write operations. The cluster endpoint provides failover support. If your cluster's current primary instance fails, the cluster endpoint automatically redirects connection requests to a new primary instance. You do not have to make changes to your application after a failover.

Reader endpoint

The reader endpoint load balances read-only connections across all available replicas in your cluster including the primary instance. When you add a replica instance to your Amazon DocumentDB cluster, it is made available for load balancing read connections using the reader endpoint. This means that you do not have to make any application changes while adding or removing read replicas in your cluster.

Instance endpoint

You can also connect to any instance in your cluster using the instance endpoint. The recommended way to connect to your cluster is to use the cluster endpoint for read/write operations and the reader endpoint for read operations. However, there may be scenarios where you create a larger than normal read replica for running analytic workloads. You can use the instance endpoint to connect and run those analytical queries against the larger instance without affecting other instances in the cluster.

Refer to the Amazon DocumentDB documentation for step-by-step instructions on creating an Amazon DocumentDB cluster and connecting to it.

Conclusion

Amazon DocumentDB is a fast, scalable, and highly available document database service that supports MongoDB workloads and is purpose-built for the cloud. It can scale to millions of requests per second and run highly scalable mission critical MongoDB workloads.

Amazon DocumentDB is a fully managed service. You do not need to worry about database management tasks, such as hardware provisioning, patching, setup, configuration, or backups. This frees you from time-consuming administration tasks and lets you focus on building your applications.

Contributors

Contributors to this document include:

- Ashok Sundaram, Solutions Architect, Amazon Web Services
- Cody Allen, Senior Solutions Architect, Amazon Web Services
- Gururaj Bayari, Senior Solutions Architect, Amazon Web Services

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Whitepaper first published	June 23, 2021
Initial publication	Whitepaper first published	May 9, 2019

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.