StreamShift® MongoDB and Azure CosmosDB API for MongoDB Integration Functional Specification

Authors	Contributors	Approved
Boopathy Vel Sanjana Mahesh	Vignesh Selvaraj Vighnesh Pai Ganesh Bushnam Shikhar Nahar	No

Revision History

Date	Version	Description	Author(s)
11-Feb-2022	1	Initial version added	Boopathy Vel Sanjana Mahesh

Overview

- MongoDB is a document storage database which stores data in flexible, JSON-like documents, which is key-value-based, supporting many levels of nesting and arrays.
- MongoDB is free and open-source, published under the GNU Affero General Public License.
- Azure Cosmos DB for Mongo API is built on top of Azure Cosmos DB, with the interactions and APIs being that of MongoDB. It supports MongoDB drivers, SDKs and tools by allowing connections to the database via connection strings.
- This document covers the integration of MongoDB as a new supported source database and Azure Cosmos DB for Mongo API as a new supported target database for Streamshift.

Scope of support in Streamshift

1. MongoDB is capable of being highly distributed, with high availability capabilities by means of sharding, i.e., by horizontally dividing large datasets into smaller units called shards.
   Therefore, a logical union across all shards is necessary to be able to obtain the entire dataset of a sharded database/collection.
   MongoDB administers sharding on the basis of shard clusters, which are individually managed by a central configuration system, query router (mongos), and the actual shard, which may be duplicated across multiple replica sets.
   For the purposes of the initial feature of integrating MongoDB into Streamshift as a source, users will be required to create a new Streamshift project for each MongoDB shard for migration to Azure Cosmos DB.
2. Since MongoDB is not a structured storage system, it is not possible to migrate to targets that are structured like RDBMS. Therefore, migrations with MongoDB as source can only be performed on similar unstructured targets - Azure Cosmos DB Sql API and Azure Cosmos DB for Mongo API
3. For the initial drop, Initial Load Only and Initial Load with CDC are only supported. CDC only mode will be supported for both Azure Cosmos targets in subsequent releases

Phases

1. Initial Configuration phase

a. Migration Type

i. Ongoing Sync Only mode is not supported for MongoDB as source

b. Source Selection

i. A new tile for MongoDB will be added under the sources pane
ii. If MongoDB is selected as source, the user is required to enter the following details -
    1. A list of endpoints and ports for each replica set
    2. Authentication type (one from No Authentication, Plain(LDAP), SCRAMSHA1, MONGODBCR, GSSAPI, MONGODBX509)
    3. Username and password
    4. Authentication database
    5. If SSL needs to be enabled to establish connection
iii. Source connection validation will be performed

c. Target Selection

i. A new tile for Azure Cosmos DB for Mongo API will be added under the Azure group for targets
ii. If MongoDB has been selected as the source, only one from the two Azure Cosmos DBs can be selected as the target
iii. If Azure Cosmos DB for Mongo API is selected as the target, the user is required to enter the following details -
    1. The hostname of the Azure Cosmos Mongo instance
    2. The port of the instance
    3. Username and password
    4. If SSL needs to be enabled to establish connection
iv. Target connection validation will be performed

d. Migration checks

i. For MongoDB source, the below checks are performed -
    1. Version must be between 2.6 and 4.0
    2. Connection to the source DB must be successful
    3. If CDC is enabled -
        a. Replication must be enabled
        b. Oplog read permission must be granted

ii. For Azure Mongo Cosmos DB, the below checks are performed -
    1. Version must be between 3.6 and 4.0
    2. Connection to the target DB must be successful

2. Assessment phase

a. Database/Schema selection

i. A list of databases found on the source machine is listed
ii. User selects the database(s) to proceed with assessment steps

b. Source Assessment

MongoDB is a NoSQL database, that supports storage of documents in a loosely typed JSON-like format, that has a nested key-value structure, with array-like constructs. Due to its loose typing, most RDBMS concepts like columns, data types, keys (primary, unique, foreign) are not present.

MongoDB has the concept of a unique *_id* key that acts as a new identifier for each record.

In addition to a document containing multiple collections, each collection can have multiple indices defined on a combination of keys.

i. Each MongoDB database is evaluated on
    1. its size,
    2. number of collections,
    3. and the total number of records it contains
ii. The migration configuration and topology also contribute to the overall scoring during the source assessment

c. Migration Compatibility

Cosmos DB for Mongo API is also a NoSQL database. The collections in the MongoDB source database can be migrated with no change to the record structure.

Each MongoDB database's compatibility is evaluated on the name of the collections at source to see if they contain unsupported characters.

There will be no SQL generation as CosmosDB only deals with native APIs to create and manage collections and databases

d. Assessment reports

i. <TBD>

3. Customization phase

a. Map Databases

i. The Map Database screen has an option to edit the target database name

ii. For now, databases containing the below special characters will not be migrated as they are either not allowed or cause errors in Azure Cosmos targets -

1. Pound symbol (#)
2. Equals sign (#)
3. Question mark character (?)
4. Percent character (%)
5. Ampersand (&)
6. Forward slash (/)
7. Backslash (\)
8. Names ending with one or more white space characters

Users will have to rename such databases prior to migration

iii. <Throughput details>

b. Customize collections

i. The customize collection page has an option to exclude and include collections into the migration

ii. Since collection creation is not SQL driven, the Edit SQL option has been disabled for the migration to target

iii. The Exclude Column option is neither applicable for the source nor target so that has been disabled for the migration to target

iv. Currently, only collections can be customized and migrated. Support for other database objects like indices will be added incrementally

v. The below fields are editable by the user -

1. **Throughput**
2. **Shard key** -

   Currently, as the user creates a project per shard, we will not be able to auto-fetch and populate the shard key to match the source. The user has to manually add the shard key(s) in this editable text field
3. **Document ID** -

   The default document id is set to the *_id* field present in all MongoDB records. This is also an editable text field where the user can enter a simple or nested key for the document id

4. Database Migration phase

a. UI tabs

Since only collections are migrated for now, there will be a single tab to display the collection migration status

b. Reports

TBD

5. Data Migration phase

a. Phases

Since only collections are migrated now, there will not be pre-data load and post-data load phases to handle non-collection objects

b. UI tabs

TBD

c. Reports

TBD

6. Notes and limitations

a. MongoDB as source can be migrated only to Azure Cosmos SQL API and Azure Cosmos Mongo API
b. MongoDB On Premise will only be supported as the source
c. IL and IL+CDC topologies only have been targeted for this drop as they require schema creation support.
d. Currently, one SMS project must be created per MongoDB shard
e. Only collections and databases from MongoDB will be migrated to Cosmos DB

7. References