What’s New in MongoDB 3.6

An overview of new features and enhancements.

Adamo Tonete

Thursday, May 24th, 2018, at 12:30 PM PDT (UTC-7) / 3:30 PM EDT (UTC-4).

PERCONA

Agenda

- Change streams
- Retryable writes
- Sessions
- Causal consistency
- Schema validation
- Security
- Query enhancements
- Miscellaneous
- Q&A

About me

Adamo Tonete

- 10+ years experience on databases.
- Senior Support Engineer
- Based in São Paulo - Brazil

About the release

- 3.6 was announced on Nov 8th, 2017
- It was the first public version that demonstrated MongoDB was going to have transactions
- A significant security improvement and more than 1k tickets solved in this release
- And a lot more...

Change Stream

Change Streams

- What is ChangeStream and how should we use it?

ChangeStream is a feature that notifies application/listeners about changes that has just happened on the database.

Resumable across replica-set, shards.

Change Streams

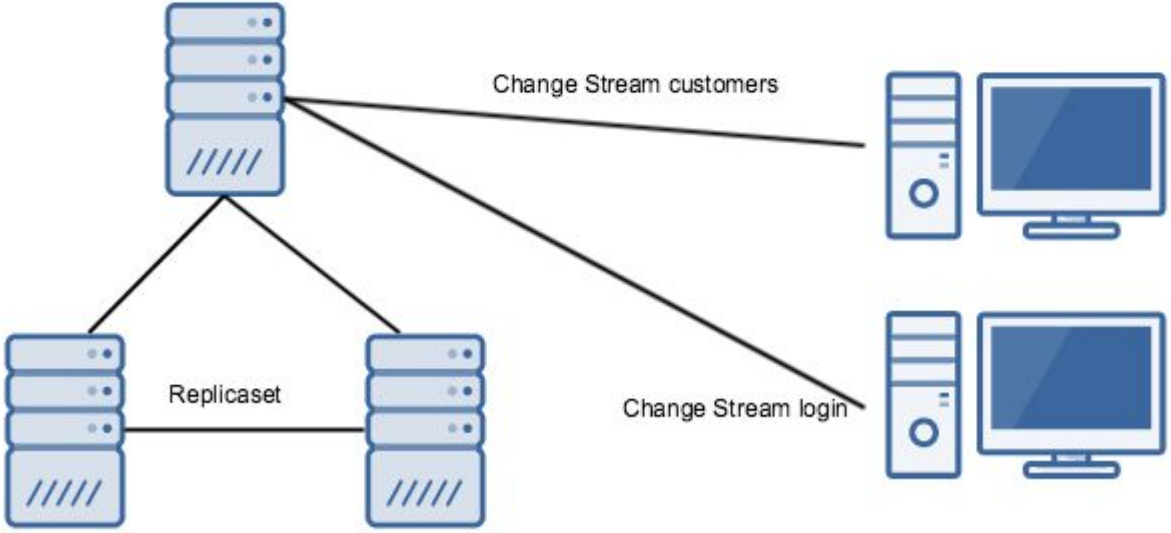
What we can captured by change stream:

- Insert
- Delete
- Replace
- Invalidate
- Update

Change Streams

```
foo:PRIMARY> db.foo.watch().pretty()
{
                   "_id" : {
                        "_data" :
BinData(0,"glsFul0AAAABRmRfaWQAZFsFul0RkbooLbup3ABaEAShVop8GcRExaRR+98mQpN1BA==")
                   },
                   "operationType" : "insert",
                   "fullDocument" : {
                        "_id" : ObjectId("5b05ba5d1191ba282dbba9dc"),
                              "x" : 2
                   },
                   "ns" : {
                        "db" : "adamo",
            "coll" : "foo"
      },
      "documentKey" : {
            "_id" : ObjectId("5b05ba5d1191ba282dbba9dc")
      }
}
```

Change Streams



Change Streams

What can be done?

Send out emails

Create metrics

Alerts

Similar to triggers

Change Streams

Downsides

It trusts writeConcern: "majority" so it will take a few milliseconds to receive the stream.

Performance is guaranteed until 1,000 streams per instance. If you need more streams, you may notice some performance degradation. However, considering you can have up to 1,000 change streams per instance, it is possible to have 3,000 change streams in a 3-nodes replica-set.

Change Streams

More info and examples:

https://www.percona.com/blog/2018/03/07/using-mongodb-3-6-change-streams/

Retryable writes

Retryable writes

- Writes may fail due to network issues, election issues, or any other hardware or communication problems.

- Retryable writes now do have transaction ids that allow the client to check whether the transaction has been applied or not.

Retryable writes

Retryable writes will try to re-apply the write when a failure occurs.

No more

```
Try
```

```
Except/catch
```

In the application side.

Retryable writes

Considerations

- Drivers must be ready for version 3.6
- MongoDB server 3.6 is required
- WriteConcern : 1 is the minimum expected
- It does not work with multi line updates/deletes

Retryable writes

More info:

https://www.percona.com/blog/2018/03/30/mongodb-3-6-retryable-writes-retryable-writes/

Sessions

Sessions

- Way to reconnect to a previous cursor state.
- Each query has its own session, we can have multiple operations in a session.
- It is possible to list all the sessions from a user and also kill process by its session.

Causal Consistency

Causal Consistency

- Operations may depend on other operations
- We can have independent and causally related operations.

What it means;

Once an operation is created, the same order will be applied for reads and writes in all the instances.

All operations must come from the same thread (session).

Schema validation

Schema validation

- Strict Document Validation
- First version of schema validation was introduced in MongoDB 3.2

It is possible now to ensure a schema for documents and if a field is required or not.

Schema validation

```
test1:PRIMARY> db.createCollection("test", {

       validator: {

              $jsonSchema: {

                     bsonType: "object",

                     required: ["x"],

                     properties: {

                            x: {

                                   bsonType: "number",

                                   description: ”field ‘x’ must be a number"

                            }

                     }

              }

       }
```

Schema validation

```
db.createCollection( "test",
   { validator: { $or:
      [
         { name: { $type: "string" } },
         { email: { $regex: /@percona\.com$/ } },
         { status: { $in: [ "Active", "Inactive" ] } }
      ]
   }
} )
```

Security

Security

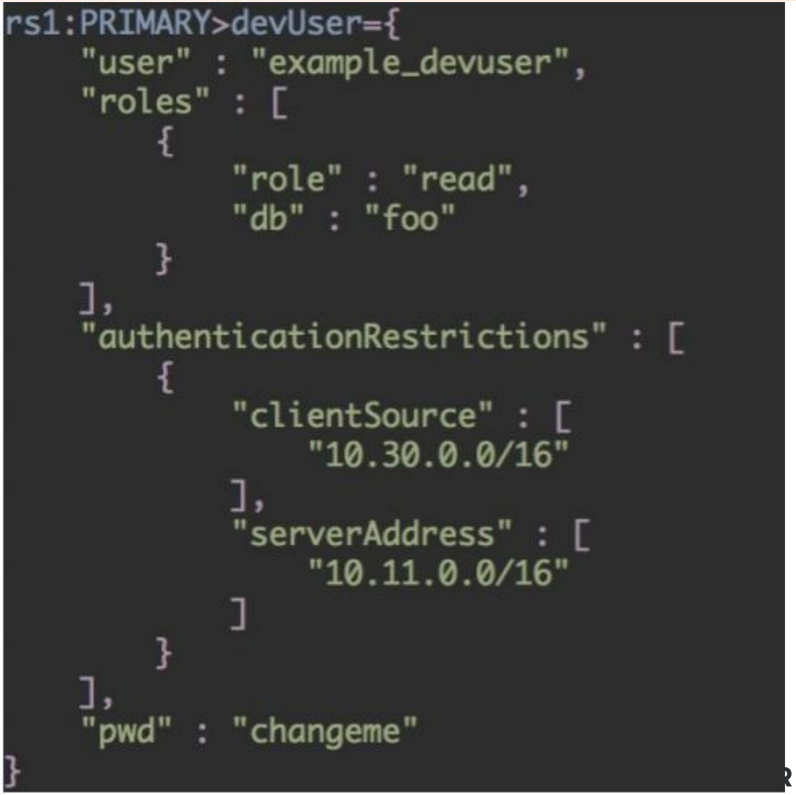
A few new improvements on security are:

- The default bindIP is now localhost instead of 0.0.0.0
- We can restrict users/group to authenticate from a specific IP.
- Similar to 'user'@'192.168.1.10' in mysql!

Security

Example of a user creation with authentication restrictions.

It is necessary to whiteList the clients ip with a CIDR block notation.



```
rs1:PRIMARY>devUser={
    "user" : "example_devuser",
    "roles" : [
        {
            "role" : "read",
            "db" : "foo"
        }
    ],
    "authenticationRestrictions" : [
        {
            "clientSource" : [
                "10.30.0.0/16"
            ],
            "serverAddress" : [
                "10.11.0.0/16"
            ]
        }
    ],
    "pwd" : "changeme"
}
```

Query enhancements

Query enhancements

Non-Aggregation

- arrayFilters
- Multi-element

Updates Arrays:

- $arrayToObject
- $objectToArray
- $mergeObjects

Changes Date

- $dateFromString
- $dateFromParts
- $dateToParts

General

- $hint
- $comment
- $$REMOVE

Query enhancements - arrayFilters

```
Input: {a: [0, 1, 3]}

db.coll.update({}, {$set: {“a.$[i]”: 2}},

   {arrayFilters: [{$or: [{i: 0}, {i: 3}]}]})

Output: {a: [2, 1, 2]}
```

Query enhancements $dateFromString

```
.. { "date": "2017-02-08T12:10:40.787" }

db.test.aggregate( [ {
   $project: {
      date: {
         $dateFromString: {
            dateString: '$date', timezone : 'America/New_York'
         }
      }
   }
} ] )

{ "date" : ISODate("2017-02-08T17:10:40.787Z") }
```

Other changes

Other changes

Change oplog size online (wiredTiger)

Intra-shard and clients can now talk to MongoD and MongoS using compression.
Drivers and MongoS use snappy compression and the advantage is that network usage goes down.

Timezone improvements

Q&A

SAVE THE DATE

OPEN SOURCE DATABASE CONFERENCE

PERCONA

LIVE EUROPE

FRANKFURT

5-7 November 2018
Radisson Blu
Frankfurt, Germany