
AWS
re:Invent

DAT211

What's new in Amazon DocumentDB (with MongoDB compatibility)

Brian Hess

Senior Specialist Solutions Architect

Amazon DocumentDB

AWS

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What's new in Amazon DocumentDB

The major new features

Other important additions

Demo



AWS re:Invent

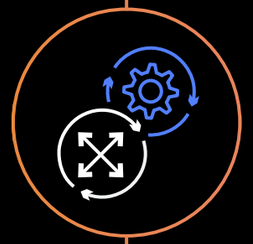
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon DocumentDB

WITH MONGODB COMPATIBILITY



Fully managed



Scalable



MongoDB compatible

Fully managed and scalable document database service that supports MongoDB workloads

Working backward from customers

December
- Canada

February
- Array operations

April
- AWS Glue read / write
- Deletion protection
- Improved multikey index

March
- Execution stats
- RBAC
- User limits

June
- T3 instances
- GovCloud
- More Amazon CloudWatch metrics
- Nulls in strings

July
- Cross-region snapshot copy

September
- Improved cursor and connection limits
- $out

August
- MongoDB 4.0 as DMS source

October
- AWS Glue crawler
- Amazon DocumentDB as DMS source
- Increased change stream retention

November
- Transactions
- MongoDB 4.0 compatibility
- São Paulo, Brazil
- Ningxia, China

Let’s start with the big stuff



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

MongoDB 4.0 compatibility

Change streams enhancements

- Longer windows (up to 7 days)
- “Start-at” capability
- Watch changes at database or cluster level

Performance improvements

RBAC enhancements

Some new operators

- $replaceRoot
- Set operations

```
# Setup
client = MongoClient(<connection details>)
db = client['mydb']
coll = db.get_collection('mycollection',
           read_preference=ReadPreference.PRIMARY)

# Create a stream object
stream1 = coll.watch()

# Read from change stream, if anything to read
event = stream1.try_next()

# Get resume token
token = stream1.resume_token

# Start stream from resume token
stream2 = coll.watch(resume_after=token)

# Start stream from set time: 2020-10-01 @noon
resdt = bson.timestamp.Timestamp(
            datetime.datetime(2020,10,1,12,0,0), 0)
stream3 = coll.watch(start_at_operation_time=resdt)
```

MongoDB 4.0 compatibility

Change streams enhancements

- Longer windows (up to 7 days)
- “Start-at” capability
- Watch changes at c

Performance impr

RBAC enhancemer

Some new operators

- $replaceRoot
- Set operations

Now with
ACID transactions

```
# Setup
client = MongoClient(<connection details>)
                                    on('mycollection',
                                    ence=ReadPreference.PRIMARY)

                                    ct


                                    eam, if anything to read
event = stream1.try_next()

# Get resume token
token = stream1.resume_token

# Start stream from resume token
stream2 = coll.watch(resume_after=token)

# Start stream from set time: 2020-10-01 @noon
resdt = bson.timestamp.Timestamp(
            datetime.datetime(2020,10,1,12,0,0), 0)
stream3 = coll.watch(start_at_operation_time=resdt)
```

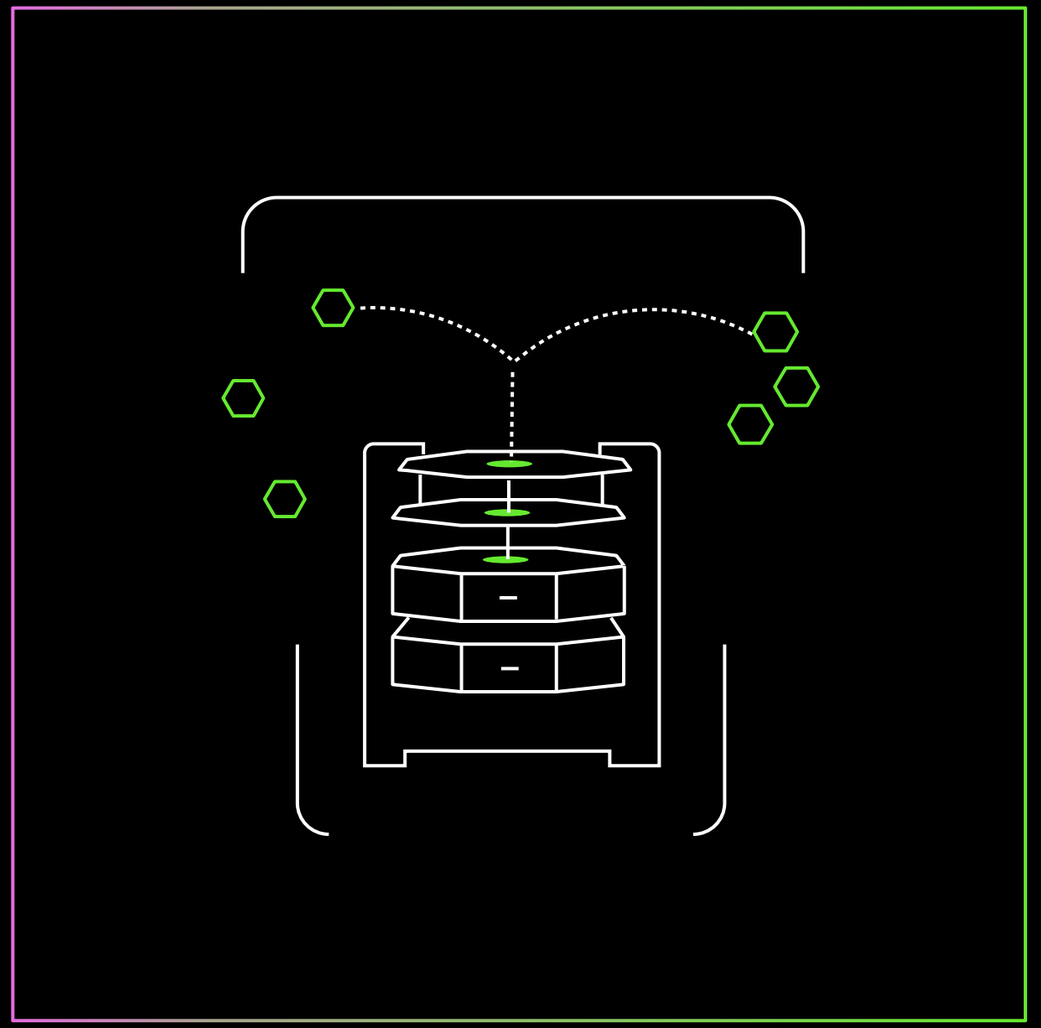
Transactions in Amazon DocumentDB

Group-write operations together

- Atomically apply all operations – ACID
- Update multiple documents in the same collection
- Update multiple collections, even in different databases
- New Amazon CloudWatch metrics to monitor transactions

Examples

- Financial transactions:
  Transfer $400 from John to Carlos
- Reservations:
  Assign a room to Jane, and update Jane's profile with the reservation



Transactions in Amazon DocumentDB

```
// Start a session
var mySession = db.getMongo().startSession();
var mySessionObject = mySession.getDatabase('test')
                               .getCollection('account');
// Start a transaction
mySession.startTransaction({readConcern: {level: 'snapshot'},
                            writeConcern: {w: 'majority'}});

// Issue writes
mySessionObject.updateOne({"_id": 1}, {"$inc": {"balance": -400}});
mySessionObject.updateOne({"_id": 2}, {"$inc": {"balance": 400}});

// Commit the transaction and end the session
mySession.commitTransaction()
mySession.endSession()
```

T3 instances

Starting at just 7.8¢ per hour

- Cheaper than a cup of coffee a day

Light-weight instance

- 2 VCPU, 4GB RAM

Perfect for development and test use cases

Partner with “stop a cluster” for significant savings

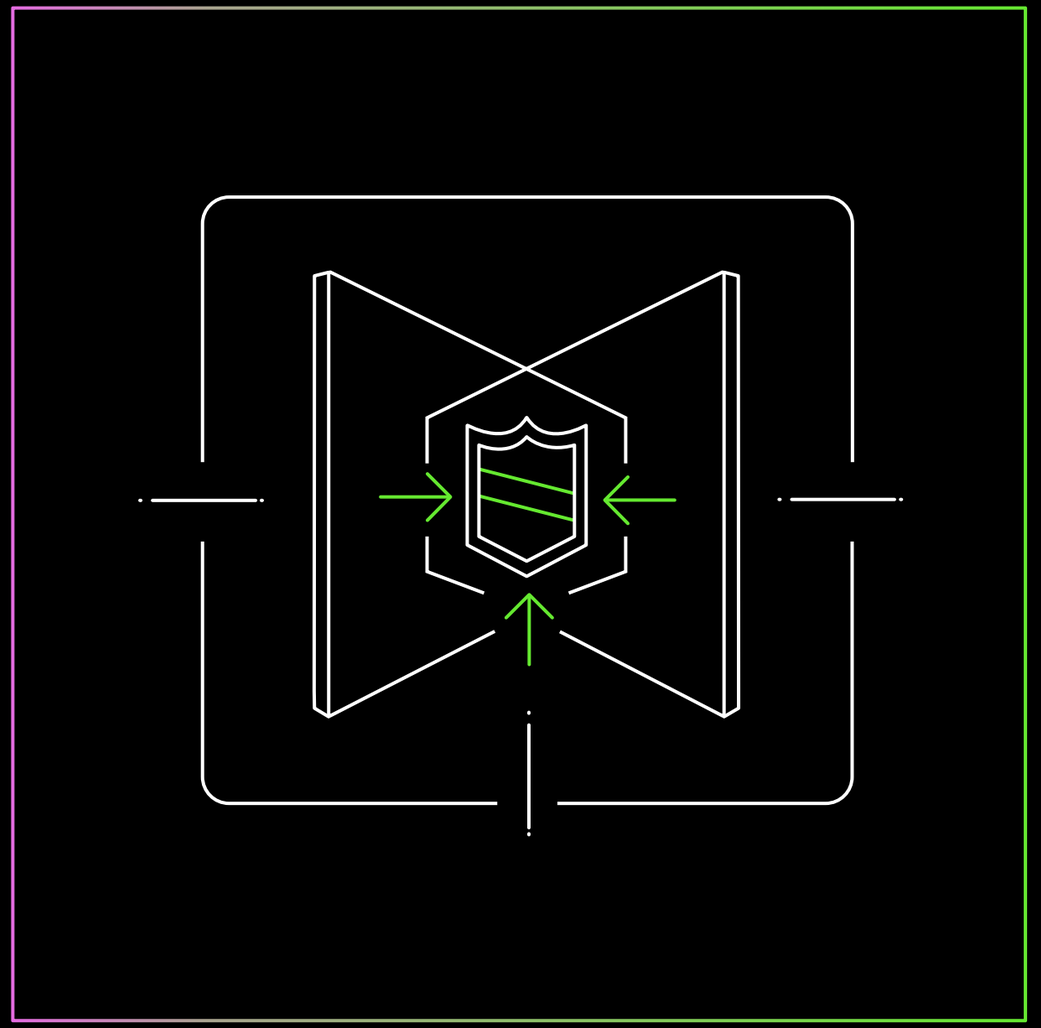
- Stop clusters overnight or on weekends
- While the cluster is stopped, pay only for existing backups and storage costs



Role-based access control

Built-in roles

- Grant access to collections in a database or all databases
  (read-only or read-write)
- Database administrator
- Cluster administrator
- Backup / restore



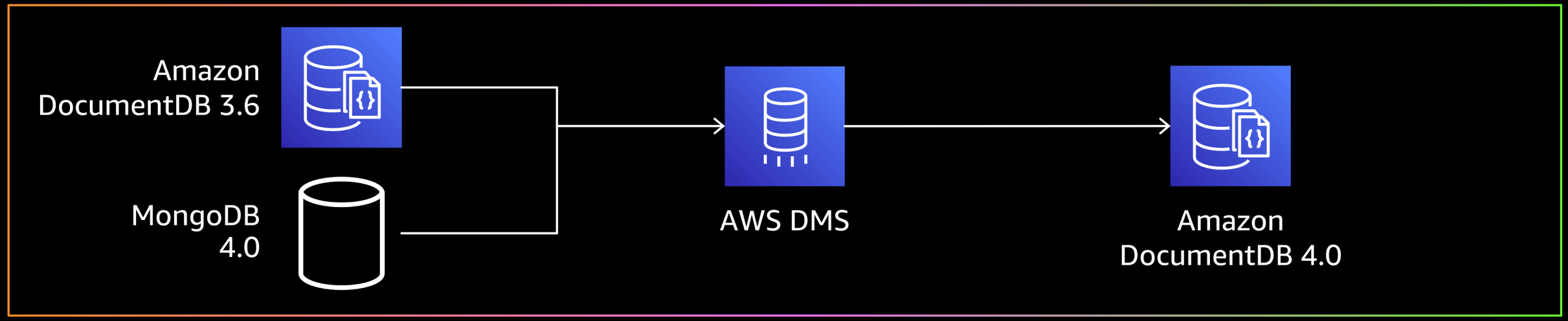
AWS Database Migration Service (AWS DMS)

Easily migrate data from MongoDB to Amazon DocumentDB

- One-time migration or ongoing replication

New this year

- Support for MongoDB 4.0 and Amazon DocumentDB 3.6 as a source
- Support for Amazon DocumentDB 4.0 as a destination
- Initial full-load performance improved by 3x



Other important additions this year



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

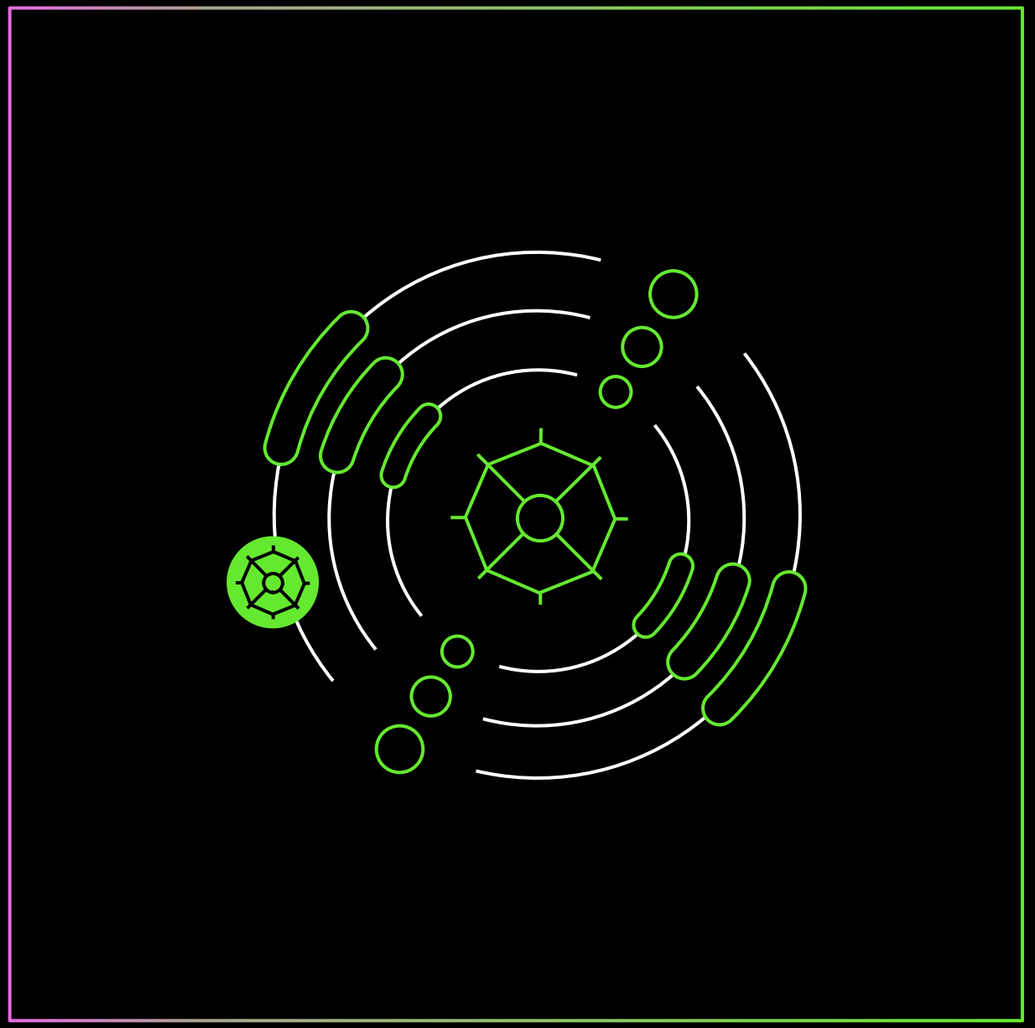
MongoDB API compatibility

Execution statistics
for explain plans

Null characters in strings

Leveraging indices for
$regex queries

Improved multi-key indexing



MongoDB API compatibility and operators



Change streams enhancements

Longer windows
(up to 7 days)

Watch changes at
database or cluster level



New operators launched

Array operations:
`$objectToArray, $arrayToObject, $slice, $range`

Arithmetic: `$mod`

Dates: `$dateFromString`

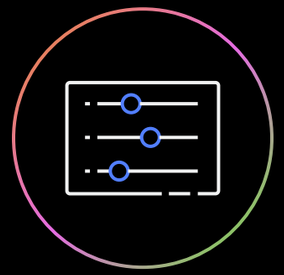
Output: `$out`

Security and resource limits



Security

- Cluster deletion protection (including AWS CloudFormation support)
- Role-based access control for built-in roles



Resource limits

- 10x increase in number of users per cluster (up to 1,000)
- Increase in cursor limit (up to 4,560) with instance class
- Increase in connection limit (up to 30,000) with instance class

New AWS Regions



Integration with AWS services



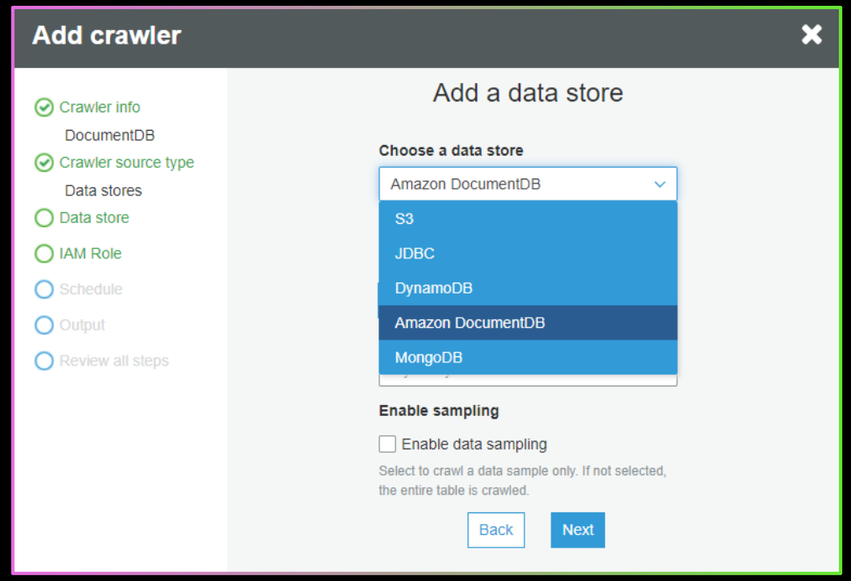
Amazon CloudWatch metrics

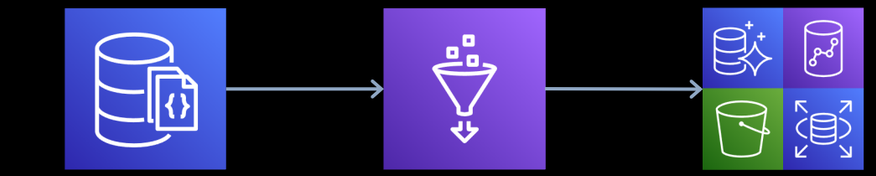
- Connections and cursors
- MongoDB op counters
- Cache hit ratio for indices
- Transaction-related metrics
- Resource governance–related metrics

Integration with AWS services

AWS Glue

- Read and write to Amazon DocumentDB
- Crawler for Amazon DocumentDB
  - Infer schema, add to catalog







Demo



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon DocumentDB v4.0 Trans | Amazon DocumentDB v4.0 Chan

demo-notebook-sagemakerinstance.notebook.us-east-2.sagemaker.aws/notebooks/Amazon%20DocumentDB%20v4.0%20Transactions.ipynb#

Apps | AWS | DocDB Docs | Amazon Document...

jupyter Amazon DocumentDB v4.0 Transactions Last Checkpoint: Yesterday at 3:07 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help | Trusted | conda_python3

Run | Markdown | nbdiff

Description

This notebook demonstrates transactions in Amazon DocumentDB. Ensure that the SageMaker Notebook instance is in a security group that can access Amazon DocumentDB.

This notebook performs the following tasks

- import some needed packages.
- gather the connection details and credentials for the Amazon DocumentDB cluster and make the connection
- issue a basic command to verify successfull connection
- issue some insert statements
- issue some read statements
- explore transactions in Amazon DocumentDB

In [ ]:
```
# Some packages we are going to use
import pymongo
import getpass
import pandas as pd

def show(x, cols=None):
    df = None
    if cols is None:
        df = pd.DataFrame(x)
    else:
        df = pd.DataFrame(x)[cols]
    display(df.sort_values(df.columns[0]))
```

Make connection to Amazon DocumentDB

In [ ]:
```
# Prompt user for Amazon DocumentDB connection details and credentials
docdb_host = "docdb40.cluster-ckhdlas9qgia.us-east-2.docdb.amazonaws.com"
docdb_port = "27017"
username = input("Username:")
password = getpass.getpass("Password:")
```

In [ ]:
```
# Make a client connection to Amazon DocumentDB with the supplied connection detals and credentials
pem_location = "/home/ec2-user/SageMaker/rds-combined-ca-bundle.pem"
uri_str = "mongodb://{username}:{password}@{docdb_host}:{docdb_port}/?ssl=true&ssl_ca_certs={pem_location}&replicaSet=rs0&readPr
uri = uri_str.format(username=username, password=password, docdb_host=docdb_host, docdb_port=docdb_port, pem_location=pem_locati
```

Untitled - Notepad

File Edit Format View Help

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

4:54 PM
11/24/2020

Amazon DocumentDB

YEAR AT A GLANCE

December

- Canada

February

- Array operators

March

- Execution stats
- RBAC
- User limit

April

- AWS Glue read / write
- Deletion protection
- Improved multikey indexing

June

- T3 Instances
- GovCloud
- More CloudWatch metrics
- Null in strings

July

- Cross-region snapshot copy

August

- MongoDB 4.0 as an AWS DMS source

September

- Improved cursor and connection limits
- $out

October

- AWS Glue crawler
- Amazon DocumentDB as an AWS DMS source
- Increased change stream retention

November

- Transactions
- MongoDB 4.0 compatibility
- São Paulo, Brazil
- Ningxia, China

Related sessions

DAT215

Zulily drives shopping with Amazon DocumentDB and Kinesis Data Analytics

DAT307

Amazon DocumentDB (with MongoDB compatibility) deep dive

DAT308

Migrating databases to Amazon DocumentDB (with MongoDB compatibility)

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.