Guía del usuario de BPA Actualización de MongoDB v5.1

Introducción

BPA v4.1.2 incluye una actualización a MongoDB v7, ya que MongoDB v5 se acerca al final de su vida útil. Este documento proporciona detalles para actualizar la versión MongoDB en microservicios.

Actualización de versiones

Actualice las siguientes versiones del paquete en microservices:

- "mongodb": "^3.1.13" a "mongodb": "^6.8.0"
- "mangosta": "5.8.7" a "mangosta": "^8.5.1"

Amortización de variables

Las siguientes variables están obsoletas en MongoDB v7:

- ssl
- useNewUrlParser
- useUnifiedTopology
- utilizarBuscarYModificar
- utilizarCrearÍndice
- sslValidate

Reemplace ssl por tls como se muestra en el siguiente ejemplo:

```
if (process.env.MONGO_SSL == "true") {
  //options.ssl = true;
  options.tls = true;
  options.tlsAllowInvalidCertificates = true;
}
```

Eliminación de devolución de llamada

La devolución de llamada en consultas MongoDB ha quedado obsoleta en MongoDB v7. La consulta actualizada se muestra en verde a continuación.

```
deviceCredentials.find(query, ['name', 'credential_id', 'credential_type', 'description'], function (error, credentials) {
deviceCredentials.find(query, ['name', 'credential_id', 'credential_type', 'description']).then(credentials => {
    let result = [];
```

Consulta actualizada

Actualización de la consulta de eliminación

Después de realizar la consulta de eliminación, la variable de respuesta debe utilizar deleteCount en lugar de n (por ejemplo, reemplace result.n por result.deleteCount).

```
if (!result.n) {
if (!result.deletedCount) {
  return res.status(404).json(errorHandlerObject.errorHandler(false, false, errorString, 404)).end('');
}
```

Reemplazar resultado.n

Actualización de cambios de consulta

Después de realizar la operación de actualización, la variable de respuesta debe utilizar modifiedCount en lugar de n/nModified (por ejemplo, Replace devices.n / devices.nModified to devices.modifiedCount como se muestra a continuación).

```
if (devices && devices.n && devices.n > 0) {
if (devices && devices.modifiedCount && devices.modifiedCount > 0) {
  response.success.push({ name: element.name, message: UPDATE_ASSETS.SUCCESS });
} else {
```

Sustituya device.n o devices.nModified.

Actualización de la función db.addUser()

Reemplace la función db.addUser() por db.command(createUse: "username") como se muestra en el siguiente ejemplo.

```
migration_db.addUser(process.env.MONGODB_USER, process.env.MONGODB_PASSWORD, {
  roles: [{
    role: 'readWrite',
    db: database_name
  }]
migration_db.command({ // in mongodb 6.x addUser function removed
  createUser: process.env.MONGODB_USER,
  pwd: process.env.MONGODB_PASSWORD,
  roles: [ { role: 'readWrite', db: database_name } ]
}). then( (result) => {
```

Reemplazar db.addUser()

Actualización de métodos

Actualice los siguientes métodos obsoletos en MongoDB v7 como se indica a continuación.

Método obsoleto	Nuevo método
actualizar	updateOne o updateMany
quitar	delete or deleteMany
cuenta	countDocuments
buscarUnoYQuitar	findOneAndDelete

Actualización de GridFS

En MongoDB v7, la clase GridFSBucket se utiliza para interactuar con GridFS, una especificación para almacenar y recuperar archivos de gran tamaño. La clase GridFSBucket proporciona métodos para cargar, descargar y administrar archivos en GridFS.

Reemplace gridfs-stream importando GridFSBucket de MongoDB. Consulte las imágenes siguientes para ver los cambios.

```
const Gridfs = require('gridfs-stream');
const { GridFSBucket } = require('mongodb');
```

Reemplazar gridfs-stream

```
let dbConn = mongoose.connection.db;
let dbDriver = mongoose.mongo;
let gridFs = new Gridfs(dbConn, dbDriver);
let readStream = gridFs.createReadStream({ _id: pdfrefId });
let gridFs = new GridFSBucket(dbConn);
let fileId = new mongoose.Types.ObjectId(pdfrefId);
let readStream = gridFs.openDownloadStream(fileId);
```

Ejemplo 1

```
let dbConn = mongoose.connection.db;
let dbDriver = mongoose.mongo;
let gridFs = new Gridfs(dbConn, dbDriver);
let gridFs = new GridFSBucket(dbConn);

let writeStream = gridFs.createWriteStream({
    filename: fileName,
    mode: 'w',
    content_type: contentType
let writeStream = gridFs.openUploadStream(fileName, {
    contentType: contentType
});
fs.createReadStream(fileName).pipe(writeStream);
writeStream.on('close', async (file) => {

writeStream.on('finish', async (file) => {

if (storeType === 'pdf') {
    reportObj.pdfrefId = file._id;
    reportObj.pdfrefId = writeStream.id;
    reportObj.pdfGenerationState = PDF_GENERATE_STATE.COMPLETED;
} else reportObj.htmlrefId = file._id;
} else reportObj.htmlrefId = writeStream.id;

});
```

Ejemplo 2

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).