

Plan de Respaldo

Sistema MotionParts - MongoDB

Introducción

Este manual describe el proceso completo para implementar un plan de respaldos para la base de datos MongoDB del sistema MotionParts. El plan incluye:

- Respaldos automáticos mediante `mongodump`
- Restauración de datos mediante `mongorestore`
- Programación de respaldos periódicos
- Gestión del ciclo de vida de los respaldos

Herramientas principales:

- `mongodump`: Crea respaldos completos de la base de datos
 - `mongorestore`: Restaura datos desde respaldos existentes
-

Instalación de MongoDB Database Tools

Las herramientas `mongodump` y `mongorestore` forman parte del paquete MongoDB Database Tools, que debe instalarse por separado.

Instalación en Ubuntu 24.04

Las herramientas `mongodump` y `mongorestore` forman parte del paquete MongoDB Database Tools, que debe instalarse por separado del servidor MongoDB.

Requisitos previos

Requisito	Comando de verificación
-----------	-------------------------

Ubuntu 24.04 (Noble)	<code>lsb_release -a</code>
----------------------	-----------------------------

Arquitectura x86_64 o ARM64 `uname -m`

cURL instalado curl --version

Nota: Si cURL no está instalado: sudo apt update && sudo apt install -y curl

Paso 1: Identificar la arquitectura del sistema

uname -m

Resultado Arquitectura Paquete a descargar

x86_64 Intel/AMD 64-bit mongodb-database-tools-ubuntu2404-x86_64-10.0.13.0.deb

Paso 2: Descargar MongoDB Database Tools

Visitar la página oficial para obtener la URL más reciente:
<https://www.mongodb.com/try/download/database-tools>

Para sistemas x86_64 (Intel/AMD):

```
cd /home/ubuntu && curl -LO  
https://fastdl.mongodb.org/tools/db/mongodb-database-tools-ubuntu2404-x86_64-100.13.0.d  
eb
```

Paso 3: Verificar la descarga

```
ls -lh /home/ubuntu/mongodb-database-tools-*.deb
```

Salida esperada (aproximadamente 50-60 MB):

```
-rw-rw-r-- 1 ubuntu ubuntu 53M dic  3 10:00  
mongodb-database-tools-ubuntu2404-x86_64-100.13.0.deb
```

Paso 4: Instalar el paquete

```
sudo apt install -y ./mongodb-database-tools-*.deb
```

Paso 5: Verificar la instalación

```
mongodump --version
```

Salida esperada:

```
mongodump version: 100.13.0
```

```
git version: abc123...
```

```
Go version: go1.x.x
```

Verificar que el paquete está registrado en el sistema:

```
dpkg -l | grep mongodb-database-tools
```

Instalación en Windows

Paso 1: Descargar el Instalador

1. Visite: <https://www.mongodb.com/try/download/database-tools>
2. Seleccione:
 - Platform: Windows x64
 - Package: ZIP
3. Descargue el archivo

Paso 2: Extraer el Archivo ZIP

Extraiga el contenido en una ubicación permanente, por ejemplo:

1. C:\Program Files\MongoDB\Tools\

Paso 3: Agregar al PATH de Windows

1. Abra Configuración del Sistema → Variables de entorno
2. En Variables del sistema, seleccione **Path** y haga clic en Editar
3. Agregue la ruta donde extrajo los binarios: C:\Program Files\MongoDB\Tools\bin
4. Haga clic en Aceptar para guardar

Paso 4: Verificar la Instalación

Abra un Command Prompt o PowerShell nuevo y ejecute:

```
mongodump --version
```

```
mongorestore --version
```

Configuración de Scripts de Respaldo

Scripts para Ubuntu 24.04

Los scripts de respaldo y restauración automatizarán el proceso y facilitarán la gestión de respaldos.

Script de Respaldo ([backup.sh](#))

```
#!/usr/bin/env bash
set -euo pipefail

# Backup script that stores the archive in the project `backup` folder
# on the host.

# Usage: ./backup.sh [output-file]

# Default output-file: backup.motionparts

HOST_BACKUP_DIR="/home/ubuntu/Motionparts/MotionParts/backup"

mkdir -p "$HOST_BACKUP_DIR"

BACKUP_FILE="${1:-backup.motionparts}"

echo "Creating backup archive: $HOST_BACKUP_DIR/$BACKUP_FILE"

# Use streaming to avoid permission issues when the container attempts
# to write into the mounted

# host directory. The container writes the archive to stdout and the
# host shell saves it.

docker run --rm --network motionparts_motionparts_net \
mongo:6.0 \
```

```

mongodump --uri='mongodb://motionparts_mongo:27017/motionparts'
--archive --gzip > "$HOST_BACKUP_DIR/$BACKUP_FILE"

if [ -f "$HOST_BACKUP_DIR/$BACKUP_FILE" ]; then

    echo "Backup written to: $HOST_BACKUP_DIR/$BACKUP_FILE"

else

    echo "ERROR: backup file not created: $HOST_BACKUP_DIR/$BACKUP_FILE"
>&2

    exit 1

fi

```

Instrucciones de uso:

Crear el archivo:

1. touch [backup.sh](#) && nano backup.sh
2. Pegar el contenido del script

Dar permisos de ejecución:

3. chmod +x backup.sh

Ejecutar el script:

4. ./backup.sh
-

Script de Restauración (restore.sh)

```

#!/usr/bin/env bash

set -euo pipefail


# Restore script that reads the archive from the project `backup`
# folder on the host.

# Usage: ./restore.sh [backup-file] [target-db]

# Default backup-file: backup.motionparts

```

```

# Default target-db: motionparts_test

HOST_BACKUP_DIR="/home/ubuntu/Motionparts/MotionParts/backup"
BACKUP_FILE="${1:-backup.motionparts}"
TARGET_DB="${2:-motionparts_test}"

echo "Restoring $HOST_BACKUP_DIR/$BACKUP_FILE into database:
$TARGET_DB"

# Use streaming: feed the archive from the host into mongorestore stdin
# inside the container.

if [ ! -f "$HOST_BACKUP_DIR/$BACKUP_FILE" ]; then
    echo "ERROR: backup file not found: $HOST_BACKUP_DIR/$BACKUP_FILE" >&2
    exit 1
fi

docker run --rm --network motionparts_motionparts_net -i \
mongo:6.0 \
mongorestore --uri='mongodb://motionparts_mongo:27017' --archive
--gzip \
--nsFrom='motionparts.*' --nsTo="$TARGET_DB.*" --verbose <
"$HOST_BACKUP_DIR/$BACKUP_FILE"

echo "Restore finished into: $TARGET_DB"

```

Instrucciones de uso:

Crear el archivo:

1. nano restore.sh
2. Pegar el contenido del script

Dar permisos de ejecución:

3. chmod +x restore.sh

Ejecutar el script:

4. ./restore.sh
-

Scripts para Windows

Script de Respaldo (backup.bat)

Cree un archivo **backup.bat** con el siguiente contenido base:

```
mongodump --uri="mongodb://localhost:27017/motionparts"  
--archive=backup.motionparts --gzip
```

1. Guarde el archivo como **backup.bat**
 2. Edite las variables según su configuración
 3. Ejecute con doble clic o desde CMD
-

Script de Restauración (restore.bat)

Cree un archivo **restore.bat** con el siguiente contenido base:

```
mongorestore --gzip --archive=backup.motionparts  
--uri="mongodb://localhost:27017/motionparts" --verbose
```

Uso:

1. Guarde el archivo como **restore.bat**
2. Edite las variables según su configuración
3. Ejecute y siga las instrucciones en pantalla

Automatización de Respaldos

Programación en Linux (Cron)

Paso 1: Editar el Crontab

```
crontab -e
```

Paso 2: Agregar Tarea Programada

Para ejecutar el respaldo diariamente a las 2:00 AM:

```
0 2 * * * /ruta/completa/backup.sh >> /var/log/mongodb_backup.log 2>&1
```

Para ejecutar cada 6 horas:

```
0 */6 * * * /ruta/completa/backup.sh >> /var/log/mongodb_backup.log 2>&1
```

Para ejecutar semanalmente (domingos a las 3:00 AM):

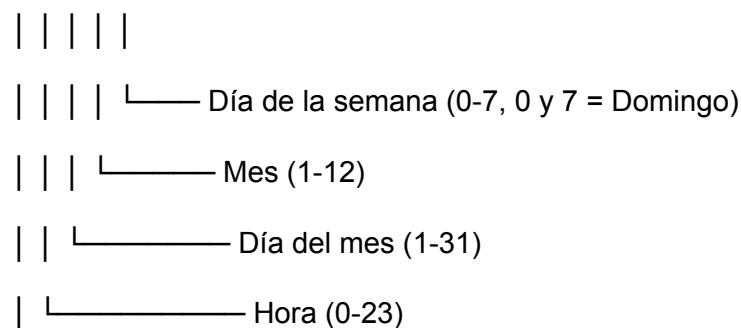
```
0 3 * * 0 /ruta/completa/backup.sh >> /var/log/mongodb_backup.log 2>&1
```

Paso 3: Verificar Tareas Programadas

```
crontab -l
```

Sintaxis de Cron

* * * * * comando



└———— Minuto (0-59)

Programación en Windows (Task Scheduler)

Método 1: Interfaz Gráfica

1. Abra Programador de tareas (Task Scheduler)
2. Haga clic en Crear tarea básica
3. Configure:
 - Nombre: Respaldo MongoDB MotionParts
 - Desencadenador: Diariamente a las 2:00 AM
 - Acción: Iniciar un programa
 - Programa: `C:\ruta\al\backup.bat`
4. Marque Ejecutar con los privilegios más altos
5. Guarde la tarea

Método 2: Línea de Comandos

```
schtasks /create /tn "MongoDB Backup" /tr "C:\ruta\al\backup.bat" /sc daily /st 02:00 /ru SYSTEM
```

Para ejecutar cada 6 horas:

```
schtasks /create /tn "MongoDB Backup" /tr "C:\ruta\al\backup.bat" /sc hourly /mo 6 /ru SYSTEM
```