This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Predictive Routing Deployment and Operations Guide

Database Maintenance

12/14/2025

Contents

Database Maintenance

This topic provides recommendations for keeping your MongoDB database up-to-date and working correctly. It includes sections covering the following:

- How to Specify a Dedicated Mount Point
- Backing Up and Restoring Your Data

Store MongoDB on a Dedicated Mount Point

MongoDB data can grow quite large depending on the size of your environment and how you configure Predictive Routing. By default, the **docker-compose.yml** file (which you obtain from Docker.com) specifies `/datadir/` as the MongoDB storage location. If you expect to write a lot of data on the MongoDB server, Genesys recommends that you create a separate mount point for that purpose.

- **Why?** If your OS uses the default configuration, the `/datadir/` directory shares storage with the root (`/`). If your Predictive Routing data files or logs grow too large, they will fill up your hard drive to a point where you might even lose access to the server.

The instructions below assume you are using Logical Volume Manager (LVM) to handle your disks. This example shows how to create a 20G logical volume for your data:

1. Create the new directory, using the following commands:

   ```
   sudo lvcreate -L 20G -n mongoVolume LVMVolGroup
    mkfs.ext4 /dev/LVMVolGroup/mongoVolume
    mkdir /new-datadir
    mount /dev/LVMVolGroup/projects /new-datadir
   ```

   To make the new mount point permanent, update the **/.../fstab** file.

2. Update those servers running MongoDB (those with the `mongo` label). To do this, open the **docker-compose.yml** file and replace the following text:

   ```
   volumes:
         - /datadir:/data/db
   ```

   with

   ```
   volumes:
         - /new-datadir:/data/db
   ```

3. (Optional) If you initialized the application by running the **start.sh** script, move the data to your new directory using the following commands:

   ```
   cd ./scripts/
    bash stop.sh
    mv /datadir/* /new-datadir/
   ```

4. Start the application:

   ```
   bash start.sh
   ```

Back Up and Restore MongoDB

This section supplies the commands needed to back up and restore MongoDB in a single-site/single-server AICS deployment. For backup and restore instructions for HA environments, see Backing Up Your Data in the Deploying: High Availability topic.

Using SSL with MongoDB

The procedure below is for MongoDB with SSL enabled. *Genesys recommends that you use SSL.*

- To use SSL, add the `--ssl` parameter to your commands.

  In test environments, you can optionally add `--sslAllowInvalidCertificates` following the `--ssl` parameter.

  In test environments ONLY, if you need to maintain an environment without SSL connections, omit the `--ssl` and `--sslAllowInvalidCertificates` parameters.

Backup MongoDB

Use the following procedure to back up MongoDB in a single-server (single node) environment:

1. Log into the container:

   ```
   docker exec -it mongo bash
   ```

2. Generate the dump of the **/backup** file:

   ```
   mongodump --ssl --out /data/db/backup --host localhost:27017
   ```

> **Tip**
>
> You can view the **/data/db/backup** directory for the container from the base system in the **/datadir/backup** directory if you are using the default defined directory for MongoDB in the **docker-compose.yml** file.

Restore MongoDB

Use the following procedure to restore MongoDB on single node installation:

1. Go to the installation directory:

   ```
   cd "IP_JOP_PRR_<version_number>_ENU_linux/scripts/"
   ```

2. Stop all the application containers:

```
bash stop.sh
```

3. Restart the MongoDB container:

```
../docker-compose -f "docker-compose.yml" up -d mongo
```

4. Start a bash session on the MongoDB container:

```
docker exec -it mongo bash
```

5. Log into the container:

```
mongo solariat_bottle --ssl --eval "db.dropDatabase()"
```

6. Restore the dump from the **/backup** file:

```
mongorestore --ssl --drop /data/db/backup --host localhost:27017
 exit
```

7. To restart all the application containers, run:

```
bash install.sh
```

and then:

```
bash start.sh
```