

AI-04469 Docmost Backup and Clean Restore

What is the bkm for Docmost Backup and Clean Restore. Create a comprehensive and simple step by step procedure. Show also a pdadmin method .

Here is the Best Known Method (BKM) for backing up and performing a clean restore of a Docmost instance.

ðŸ”” Core Components of Docmost

To make a complete backup, you must preserve three elements:

1. PostgreSQL Database: Page structures, user accounts, space configs, and permissions.
2. Uploaded Files / Data Volume: Attachments, uploaded images, and media files.
3. Environment File (.env): Configuration keys (especially APP_SECRET, database passwords, and SMTP/S3 credentials).

Method 1: Docker CLI / Command Line (Recommended BKM)

Step 1: Backup Procedure

1. Stop the App Container

Stop the Docmost application container while leaving the PostgreSQL database container running to prevent data writes during the backup.

```
docker stop docmost
```

2. Dump the PostgreSQL Database

Run pg_dump from inside your PostgreSQL container:

```
docker exec -t docmost-db pg_dump -U docmost_user -d docmost_db > docmost_db_backup.sql
```

(Replace docmost-db, docmost_user, and docmost_db with your actual container name and credentials defined in your .env).

3. Backup Files & Environment Config

Compress the uploaded assets directory or named Docker volume along with your configuration file:

```
# Backup docker volume/data folder
```

```
tar -czvf docmost_data_backup.tar.gz /path/to/docmost/data
```

```
# Backup environment file
```

```
cp .env docmost.env.backup
```

4. Restart the App Container

```
docker start docmost
```

Step 2: Clean Restore Procedure

1. Prepare Target Environment

Ensure Docker is installed and copy your `docmost.env.backup`, `docmost_db_backup.sql`, and `docmost_data_backup.tar.gz` to the target server.

2. Start PostgreSQL Container Only

Run PostgreSQL first so it can receive the restored data:

```
docker compose up -d db
```

3. Drop & Recreate standard Database (Clean Wipe)

To guarantee a clean restore with no conflicting schema tables, drop and recreate the database:

```
# Drop existing DB
```

```
docker exec -i docmost-db dropdb -U docmost_user docmost_db
```

```
# Recreate empty DB
```

```
docker exec -i docmost-db createdb -U docmost_user docmost_db
```

4. Restore PostgreSQL Database

Feed the backup file into the freshly created database:

```
docker exec -i docmost-db psql -U docmost_user -d docmost_db < docmost_db_backup.sql
```

5. Restore File Attachments & Config

Extract your user uploads back into the Docmost data path/volume and restore `.env`:

```
# Extract user uploads
```

```
tar -xzf docmost_data_backup.tar.gz -C /path/to/docmost/data
```

```
# Restore environment file
```

```
cp docmost.env.backup .env
```

6. Start All Docmost Services

```
docker compose up -d
```

Method 2: pgAdmin Method (GUI-Based)

If you manage PostgreSQL through pgAdmin, you can execute database backups and clean restores directly through the UI.

Note: File attachments must still be backed up/restored at the folder/volume level on the server (Step 3 & Step 5 of Method 1).

Step 1: Connect pgAdmin to Docmost DB

1. Open pgAdmin and right-click Servers > Register > Server...
2. In the General tab, set a name (e.g., Docmost DB).
3. In the Connection tab:
 - Host name/address: localhost or container IP/hostname
 - Port: 5432
 - Maintenance database: docmost_db
 - Username: docmost_user
 - Password: (Your PostgreSQL password)
4. Click Save.

Step 2: Backup via pgAdmin

1. Expand Servers > Docmost DB > Databases.
2. Right-click on docmost_db and select Backup...
3. In the dialog:
 - Filename: docmost_backup.tar (or .sql / .backup)
 - Format: Select Custom or Tar (recommended)
 - Encoding: UTF8
4. Click the Backup button. pgAdmin will generate the backup file on your client machine.

Step 3: Clean Restore via pgAdmin

1. Disconnect Active Connections

Before dropping/restoring, ensure the Docmost container is stopped so there are no active locks:

```
docker stop docmost
```

2. Drop and Recreate the Database in pgAdmin

1. In pgAdmin, right-click docmost_db > Delete/Drop.
2. Right-click Databases > Create > Database...

3. Name it docmost_db and set the owner to docmost_user.

3. Restore Database

1. Right-click the newly created docmost_db > Restore...

2. In the General tab:

- Format: Select Custom or tar (matching your backup format)
- Filename: Select your docmost_backup.tar file.

3. Click Restore.

4. Restart Docmost Container

Once pgAdmin shows Process Completed, start Docmost:

```
docker start docmost
```

â€” Important Best Practices

- Always retain APP_SECRET: Keep the exact same APP_SECRET value in your .env file across restores. If changed, encrypted data (such as API tokens or integration secrets) won't decrypt.
- Stop App Writes: Always stop or pause the docmost web service before backing up/restoring the database to avoid out-of-sync attachments and database states.
- Match PostgreSQL Versions: Ensure the target PostgreSQL version is equal to or higher than the source PostgreSQL version to prevent migration issues.