

AI-04477 Docmost and Wordpress DB Backup

How to export all my Docmost and Wordpress contents for disaster recovery from Google Cloud Platform to OnPrem. I have pgAdmin and phpMyAdmin

For a complete **Disaster Recovery (DR)** strategy, backing up database tables alone is not enough — you need a pair consisting of **(1) The Database Export** and **(2) The Application Files & Environment Configurations**.

Here is the step-by-step guide to exporting both systems from Google Cloud Platform (GCP) to your local On-Premises environment.

1. WordPress Export

WordPress relies on a **MySQL/MariaDB** database and a file directory storing media, plugins, and themes.

Step A: Export the Database (via phpMyAdmin)

1. Log in to your **phpMyAdmin** instance running on GCP.
2. Select your WordPress database from the left-hand sidebar (e.g., `wordpress` or `wp_db`).
3. Click the **Export** tab at the top.
4. Select **Quick** export method and **SQL** format (or choose *Custom* if you want to enable GZIP compression for a large database).
5. Click **Export** to download the `.sql` file to your computer.

CLI Alternative (`mysqldump`): If phpMyAdmin times out on a large database, run this on your GCP VM:

Bash

```
mysqldump -u wp_user -p wordpress_db | gzip > wordpress_backup.sql.gz
```

Step B: Export WordPress Files

You must back up the `wp-content` directory (contains uploads, themes, plugins) and your `wp-config.php` file:

1. Connect to your GCP VM via SSH.

2. Navigate to your WordPress root folder (often `/var/www/html` or `/srv/www/wordpress`).
3. Compress the essential files into an archive:

Bash

```
tar -czvf wordpress_files.tar.gz wp-content wp-config.php .htaccess
```

2. Docmost Export

Docmost relies on a **PostgreSQL** database and local file/attachment volumes (or an S3/GCS bucket).

Step A: Export the Database (via pgAdmin)

1. Log in to **pgAdmin**.
2. Right-click your Docmost database in the left panel (e.g., `docmost`) and select **Backup...**
3. Fill in the options:
 - **Filename:** `docmost_db_backup.sql` (or `.dump`).
 - **Format:** Select **Plain** (text/SQL format) or **Custom** (for `pg_restore`).
4. Click **Backup** to generate the file, then download it locally through pgAdmin's Storage Manager or browser prompt.

CLI Alternative (`pg_dump`): If running Docmost in Docker on GCP, execute directly from the server:

Bash

```
docker exec docmost-db pg_dump -U docmost_user -d docmost > docmost_database.sql
```

Step B: Export Docmost Files & Configurations

Docmost stores attached documents and images in a volume or local storage directory, along with critical environment settings.

1. SSH into your GCP VM.
2. Locate your Docmost directory (where `docker-compose.yml` and `.env` live).
3. Archive your configuration files and uploaded data volume:

- **If using Docker Volumes:**

Bash

```
docker run --rm -v docmost_data:/source -v $(pwd):/backup alpine tar czf
/backup/docmost_storage.tar.gz -C /source .
```

- **If using local directory mounts:**

Bash

```
tar -czvf docmost_files.tar.gz docker-compose.yml .env ./data
```

3. Transferring Files from GCP to On-Premises

Once the archives are created on your GCP VM, transfer them directly to your On-Premises server or local machine using one of these methods:

Option 1: Google Cloud SDK (`gcloud compute scp`)

Run this command from your local terminal on-premise:

Bash

```
gcloud compute scp --recurse username@gcp-instance-name:/path/to/backups/ ~/onprem-backups/ --zone=your-instance-zone
```

Option 2: Standard SCP or SFTP

If SSH keys are configured on your GCP VM:

Bash

```
scp user@gcp-vm-ip:/path/to/wordpress_files.tar.gz ~/onprem-backups/  
scp user@gcp-vm-ip:/path/to/docmost_storage.tar.gz ~/onprem-backups/
```

4. Disaster Recovery Checklist Summary

To guarantee full restoration on-premises, make sure your DR bundle contains these **4 core items**:

Application	Database Export	Application Files / Attachments	Config Files
WordPress	.sql file via phpMyAdmin	wp-content/ directory	wp-config.php

Docmost	.sql / .dump file via pgAdmin	Docmost upload data/storage folder	docker-compose.yml & .env
----------------	----------------------------------	---------------------------------------	------------------------------