eOTA Data Persistence/backup using Docker Volumes

eOTA leverages Docker volumes to ensure persistent storage of data. All uploaded OTA packages and their associated metadata are automatically retained, even across container restarts or recreations.

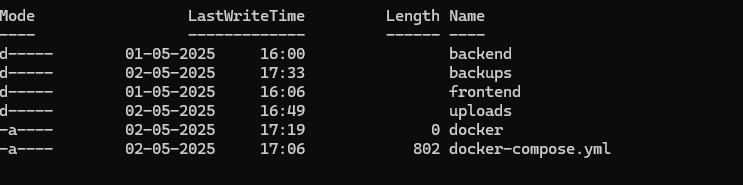
What Are Docker Volumes?

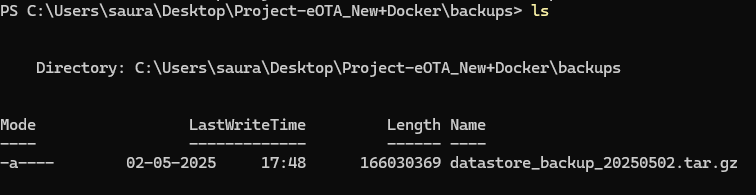
Docker volumes provide a reliable mechanism for persisting data independently of the container lifecycle. Since volumes exist outside the scope of containers, the data remains intact even when containers are stopped or removed. Docker manages these volumes and stores them in a dedicated directory on the host system, typically located at /var/lib/docker/volumes on Linux.

By utilizing Docker Compose, eOTA ensures that data is automatically preserved on the host machine. However, if users require access to this data outside the host system or wish to create backups, they can do so using the following procedure:

1. **Creating backup:**

List the present volume by using command:

1. docker volume ls (A Docker volume will always be present, as it is automatically created by our launch script(compose yaml) during project initialization.)
   1. 
2. Make a folder named “backups” inside your project root – mkdir backups
   1. 
3. Create tar of the volume: docker run --rm -v <project-eota\_newdocker\_datastore>:/source -v ${PWD}/backups:/backup alpine sh -c "tar -czvf /backup/datastore\_backup\_$(Get-Date -Format 'yyyyMMdd').tar.gz -C /source ."
   1. Note: volume name “project-eota\_newdocker\_datastore” should be same listed on “docker volume ls “.
   2. It may sometime have a project prefix as shown below: **project prefix** (e.g., <project-eota\_newdocker>)
   3. Additionally: 'yyyyMMdd' in command can be modified 'yyyyMMdd\_HHmm' to include hours and minutes of backup.
4. Back up will be created as follows with the date of backup in folder backups with tar.gz format:



1. **Restoring Backup Volumes**  
   To restore a backup, extract the contents of the .tar file and place them inside the uploads folder located in the backend directory. If the uploads folder does not exist, create it manually. Once the files are in place, start the project using docker compose up with the appropriate YAML file. Upon startup, the application will recognize the restored data, and all contents from the previous snapshot should be available.