

How to Restore a Postgres Dump

There is an automated job on the cluster that dumps the Postgres database to a plain-text sql file and saves it to a Cloud Storage bucket every day.

In order to restore the dump, here is one possible line of commands to do it:

- i. Grab the encrypted backup (`'file.dump.encrypted'`) from the Cloud Storage bucket.
- ii. Decrypt the GPG key from the backend-database-backup service: `gcloud kms decrypt --ciphertext-file ./services/backend-database-backup/.env.encrypted --plaintext-file backup-secrets --location us-east1 --keyring ufincs2-key-ring --key ufincs2-key`
- iii. Modify the decrypted backup-secrets file to remove everything except what follows `ENCRYPTION_KEY=` so that only the key itself remains.
- iv. Decrypt the dump file: `cat backup-secrets | gpg --batch --yes --passphrase-fd 0 --decrypt --cipher-algo AES256 'file.dump.encrypted' > 'file.dump'`
- v. Copy the sql dump file to the postgres container: `kubectl cp 'file.dump' 'container':/tmp`
- vi. Run a shell on the container: `kubectl exec -it 'container' bash`
- vii. Change to the dump directory: `cd /tmp`
- viii. Delete the existing database: `dropdb -U app-database-user app-database`
- ix. Create the database fresh: `createdb -U app-database-user app-database`
- x. Restore the sql dump: `psql -U app-database-user -d app-database -f ./file.dump`

We have to delete and recreate the database because the dump file contains information on how to create the tables, indexes, constraints, etc. Since the database will have all these things from the backend-migrate service (and all these things are unique), we need to get rid of them first, and then just create an empty database to restore against.