

Title: Restoring redis running on docker from dump.rdb

Post Body:

What I want to do is to use a dump.rdb that I've taken from a production server, and use it in my development environment, that is defined by a very simple compose file.

For simplicity, assume that my app is the same as this [compose example from the docker docs](#) for redis and flask, so the docker-compose.yml looks like:

```
version: '2' services: web: build: . ports: - '5000:5000' volumes: - ./code depends_on:
```

This persists redis data between restarts, but you just cannot access the redis files as there is no volume mounted for redis in the docker-compose.yml. So I change my compose file to mount a volume for redis, and I also want to force redis to persist data and the [official redis image docs](#) say that happens if I use 'appendonly'.

```
redis: image: redis command: redis-server --appendonly yes volumes: - ./redis:/data
```

If I do this, my data are persisted, as they were in the original example, and I can now see a dump.rdb and and appendonly.aof in the /redis path. The problem is, if I want to restore from a dump.rdb I need to turn off appendonly (for example, see digital ocean's how-to-back-up-and-restore-your-redis-data-on-ubuntu-14-04), and without append-only I cannot see how to get the compose file to write to the volume.

How can I produce a docker compose that will persist redis in a volume where I can switch the dump.rdb files, and therefore insert the production snapshot into my development environment?

Update The following compose works, but be patient when testing, as the creation of the dump.rdb is not instant (hence it seeming like it failed). Also the redis official image doc, implies you have to use appendonly when you don't:

```
redis: image: redis volumes: - ./redis:/data
```

Accepted Answer: None

Highest Rated Answer:

The appendonly part is just to make sure that you don't lose data, but since you already have the dump.rdb from your server you don't need to worry about that: you can either remove the append only flag or remove 'command' entirely since it will then fall back to the image default which is just 'redis-server'.

~~I have a similar setup [here](#) and it writes/loads the dump.rdb files fine. (404)~~