Running mysqldump for daily backup from a separate Docker container

Asked 11 months ago Active 11 months ago Viewed 636 times

0

Is it a bad idea to create a separate `docker` container to run `mysqldump` using a cron job for daily backups?

Most people are using either the host machine's cron job or a separate cron container to run `mysqldump` from *inside* the container which is being backed up.

I would find it nicer to install `mysql` and execute `mysqldump` in the containder dedicated for backups. It would make the entire setup more segregated.

Are there any disadvantages or this approach?

Example of such a docker-compose-yml:

```
mysql:
  image: mysql:latest
  environment:
    MYSQL_DATABASE: mydb
    MYSQL_USER: myuser
    MYSQL_PASSWORD: mypassword
    MYSQL_RANDOM_ROOT_PASSWORD: '1'
  volumes:
    - ./mysql/mysql-data:/var/lib/mysql

mysql-cron:
  image: mysql:latest
  build: .
  environment:
    MYSQL_HOST: mysql
    MYSQL_DATABASE: mydb
    MYSQL_USER: myuser
    MYSQL_PASSWORD: mypassword
    MYSQL_RANDOM_ROOT_PASSWORD: '1'
  volumes:
    - ./backup:/var/backup
```

The `Dockerfile` for mysql-cron would install 'cron' and setup crontab (not ready yet).

mysql docker cron

asked Aug 31 '19 at 14:35

adamsfamily
785 7 18

maybe replication instead? – Flash Thunder Aug 31 '19 at 14:40

your own image - you can run the dump on `cron` of the host (assuming linux) where the `docker` environment is running instead of building a custom image with `cron`. – masseyb Aug 31 '19 at 14:53

Yes it's true that you can exec in a running container but the significant drawback of this approach is that you need to configure cron on the *host* machine (therefore it signifacantly reduces reusability across different Linux distros and also segregates the backup script from the actual db container). That's why I think that it's more elegant to set up the mysqldump backup also inside of the docker ecosystem. Are there any drawbacks on my solution above? Thanks! – adamsfamily Sep 1 '19 at 18:22

@FlashThunder Yes replication is an interesting approach but it would take twice as much disk space in my case and also I'd need to monitor that the db replicas are in sync... I wanted to save this headache for myself. Other than that, yes, replication is a good approach. – adamsfamily Sep 1 '19 at 18:23

I'm not aware of any linux / unix distribution that doesn't have `cron` - I'm lazy, wouldn't maintain an image unless I absolutely have too (is a drawback imo). For `docker` I'd use the host's `cron` and for `kubernetes` a `CronJob` task. At the end of the day, if it works then I'd never say it's a "bad idea". – masseyb Sep 2 '19 at 9:11

1 Answer

Active Oldest Votes

1

Referring to the official mysql image's documentation you can `docker exec` into the running container to dump the database to a path on the host: `docker exec some-mysql sh -c 'exec mysqldump --all-databases -uroot -p"$MYSQL_ROOT_PASSWORD"' > /some/path/on/your/host/all-databases.sql`

Assuming that you're using `docker` on Linux, chances are that you already have `cron` installed (you don't need to build and maintain your own image with `cron`).

You can add a `cron` job on the host where `docker` is running to `exec` into the container periodically and dump the database(s) (`crontab -e`) i.e. every day at 01:00 A.M.: `0 1 * * * docker exec some-mysql sh -c 'exec mysqldump --all-databases -uroot -p"$MYSQL_ROOT_PASSWORD"' > /some/path/on/your/host/all-databases.sql`

Note: `cron` runs with a specific environment (e.g. ensure `docker` is in the `$PATH` available to `cron`, etc).

answered Aug 31 '19 at 15:01
masseyb
1,409 1 9 16