

Avoir sa base de données hébergée sur l'Internet : rdbhost

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 novembre 2009. Dernière mise à jour le 22 novembre 2009

<http://www.bortzmeyer.org/rdbhost.html>

Si on veut avoir sa base de données hébergée quelque part sur l'Internet et accessible depuis des applications qui peuvent tourner à divers endroits, on a bien sûr plusieurs solutions. Je présente ici un hébergement chez rdbhost <<http://www.rdbhost.com/>>.

Si on est administrateur système de profession, ou simplement si on a besoin d'applications très spécifiques, qu'on a peu de chances de trouver dans une offre standard, la meilleure solution est sans doute de louer une machine dédiée, par exemple chez Slicehost <<http://www.bortzmeyer.org/slicehost-debut.html>>. Mais si on est un programmeur ordinaire, qui n'a ni le temps, ni l'envie de faire de l'administration système, un service hébergé peut être intéressant. Le public typique visé par rdbhost <<http://www.rdbhost.com/>> est le client de Google App Engine qui veut une autre base de données que celle fournie par Google.

Que fournit rdbhost <<http://www.rdbhost.com/>> ? Un service gratuit (pour les usages raisonnables) de base de données, vous donnant accès à une base SQL (en l'occurrence, PostgreSQL). L'interface avec la base n'utilise pas le protocole réseau habituel de PostgreSQL, mais un protocole spécifique à rdbhost, au dessus de HTTP (et qui a donc de fortes chances de passer les pare-feux). Pour les programmeurs Python, rdbhost fournit une bibliothèque conforme au standard « Python DB » (PEP 249 <<http://www.python.org/dev/peps/pep-0249/>>).

Voici donc un exemple en Python :

```
from rdbhdb import rdbhdb
import sys

role = 'r0000000837'
authcode = 'trop secret'
conn = rdbhdb.connect(role, authcode)
conn.https = True
cur = conn.cursor()
cur.execute("INSERT INTO Test (data) VALUES ('%s')" % sys.argv[1])
cur.execute("SELECT * FROM Test")
for tuple in cur.fetchall():
    print tuple
conn.close()
```

On note le `conn.https = True`. Par défaut, la connexion n'est pas chiffrée et le mot de passe circule donc en clair. Un exemple de script plus riche est celui de chargement de la base de StackOverflow http://www.rdbhost.com/downloads/import_so_data_sep09.py.txt.

Pour créer les tables et les modifier, on peut utiliser des requêtes SQL DDL comme `CREATE TABLE` mais aussi une interface Web pratique, fournie par rdbhost.

Le service est gratuit, mais il vient avec un certain nombre de limites, comme un maximum de cent tuples en réponse à un `SELECT`. Si on veut stocker de grosses quantités de données, récupérer davantage de tuples, etc, il faut s'abonner au service payant. Et, naturellement, comme avec toute base de données hébergée, on n'a pas de garantie sur la confidentialité des données ou sur leur pérennité.

Il est donc indispensable de pouvoir sauvegarder ses données <http://www.bortzmeyer.org/sauvegarde-donnees-distantes.html>. Pour cela, rdbhost fournit un service de récupération http://www.rdbhost.com/dmp_form.html du contenu de ses bases.

Le service est encore assez récent et on peut donc donner son avis sur `Uservice` <http://rdbhost.uservice.com/>.

Il existe un certain nombre de services ayant des caractéristiques similaires (merci à Fil et à David Larlet pour leurs informations) :

- SimpleDB <http://aws.amazon.com/simplifiedb/> d'Amazon ne fournit pas d'interface SQL, et la documentation se moque du monde en prétendant qu'avec des bases de données relationnelles, on ne peut pas ajouter de colonnes a posteriori.
- RDS <http://aws.amazon.com/rds/>, d'Amazon, est par contre une vraie base relationnelle (mise en œuvre avec MySQL). Contrairement à rdbhost, on peut apparemment utiliser n'importe quel client MySQL, le protocole d'accès est celui du SGBD. RDS est payant, même pour les petites bases.
- FathomDB <http://fathomdb.com/> utilise aussi MySQL. Il n'y a guère de détails sur le site Web et le service est actuellement en beta et je n'ai pas d'invitation.

Connaissez-vous d'autres services équivalents ?