

Hobo - Development #17725

Ajouter une commande de backup

20 juillet 2017 12:41 - Jean-Baptiste Jaillet

Statut:	Rejeté	Début:	20 juillet 2017
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Sur le modèle de celle de w.c.s., essayer d'avoir les mêmes paramètres s'il y en a ou homogénéiser les deux.			
<ul style="list-style-type: none">• backup de la base postgres• tarball du répertoire du tenant			

Historique

#2 - 20 juillet 2017 13:00 - Frédéric Péters

Il y a déjà du code différent pour appeler le delete_tenant de w.c.s.; plutôt qu'ajouter du code ici, j'appellerais la commande backup qui existe déjà pour produire un .tar.gz, qu'il est évidemment possible d'étendre pour inclure également un dump sql dedans.

Mais je ne mêlerais pas archivage et suppression dans la même commande.

#3 - 20 juillet 2017 13:13 - Frédéric Péters

Pardon, je parlais de w.c.s., là. (mais pour suivre ce que fait w.c.s. avoir une commande d'archivage différente de la commande de suppression, ça me semble avoir du sens).

#4 - 20 juillet 2017 13:14 - Jean-Baptiste Jaillet

Ok donc création d'une commande d'archivage (dump et zip des répertoires dans une archive tar.gz).

#5 - 20 juillet 2017 14:48 - Benjamin Dauvergne

- *Sujet changé de Améliorer delete_tenant à Ajouter une commande de backup (sur le modèle de celle de w.c.s., essayer d'avoir les mêmes paramètres s'il y en a ou homogénéiser les deux)*

- *Description mis à jour*

#6 - 20 juillet 2017 14:48 - Benjamin Dauvergne

- *Sujet changé de Ajouter une commande de backup (sur le modèle de celle de w.c.s., essayer d'avoir les mêmes paramètres s'il y en a ou homogénéiser les deux) à Ajouter une commande de backup*

- *Description mis à jour*

#7 - 20 juillet 2017 18:30 - Jean-Baptiste Jaillet

- *Tracker changé de Bug à Development*

Hum j'ai fait l'archivage (je ne mets rien pour l'instant en l'absence de test et dans un souci de préserver votre patience).

J'ai une question pour la parti pg_dump :

Déjà pour être sûr pour les settings :

```
tenant_settings = settings.get_tenant_settings(settings.get_wrapped(), tenant) # settings importé de django.conf et tenants objet Tenant chargé depuis les args de la commande
```

Et la (j'ai un doute sur le DB['USER'] mit en commentaire du ticket #16302 de benj) j'ai accès à DATABASES['default']['HOST'] et ['USER'], mais aussi password si je dis pas de bêtise.

Du coup pourquoi passé par la variable d'environnement PGPASSWD?

Si on passe par PGPASSWD, est-ce que du coup elle est set au moment de l'appel comme on est connecté avec le tenant?

Et ensuite l'appel serait un :

```
subprocess.check_call(['pg_dump', '-oC', ...])
```

Et j'ajoute le dump dans le repertoire de backup qui contient aussi le tarball

#8 - 21 juillet 2017 19:05 - Jean-Baptiste Jaillet

- Fichier 0001-multitenant-add-backup_tenant-command-17725.patch ajouté

- Statut changé de Nouveau à Information nécessaire

- Patch proposed changé de Non à Oui

Hop, je suis aller tester sur la dev de sictiam ça marche.

Problème que j'ai avec les tests, c'est qu'il se connecte avec mon user postgres donc quand je fais un pg_dump du schema, il ne le connait pas et ça crache.

J'ai cherché un peu tous les settings et compagnie et là j'avoue je veux bien une astuce (sur la dev j'ai testé avec fargo et hobo, qui se connecte bien à leurs bases donc j'ai un dump et un tar.gz).

#9 - 22 juillet 2017 04:21 - Jean-Baptiste Jaillet

- Fichier 0001-multitenant-add-backup_tenant-command-17725.patch ajouté

Bon, j'ai modifié tout ça:

- rajouté le port (connection à distance, le port peut être différent)
- le nom de la db (et un truc un peu cheap en cas d'user 'root' qui se connecte et du coup ne trouve pas le schema puisqu'il n'est pas connecté à la base de l'appli. Donc je récupère la base par le nom du service. CF: je demandais conseil plus haut, sûrement un problème de settings des tests multitenant mais j'ai gratté tout ce que je pouvais et j'ai trouvé que ça pour passer les tests. Sinon en temps normal, on récupère le nom de la db (ce qui n'est pas obligatoire) depuis l'entrée DATABASE['default']['NAME']
- l'option -w, en relisant bien la doc dans mon patch plus haut j'avais fait ça n'importe comment. Donc je précise -w pour éviter le prompt, et en cas de pwd pour se connecter (ce qui sera le cas chez sictiam puisque connection à distance et chaque db a un pwd), je set PGPASSWORD puis une fois le dump fait j'unset la variable (on ne sait jamais, je me dis que si on la laisse traîner ça peut causer des soucis ensuite).

Du coup les tests passent, mais c'est un peu de la 'triche' :

```
+         if not db_name:
+             db_name = tenant.get_service()['service-id'].replace('-', '_')
```

Donc encore une fois, je ne sais pas comment sont paramétrés les tests mais depuis le tox il me manque un truc (

```
tox -e multitenant -- tests_multitenant/test_tenant_command.py
```

depuis la racine de l'app django)

C'est un peu tard donc j'espère que ce que j'écris est compréhensible.

#10 - 22 juillet 2017 10:56 - Benjamin Dauvergne

Jean-Baptiste Jaillet a écrit :

Hum j'ai fait l'archivage (je ne mets rien pour l'instant en l'absence de test et dans un souci de préserver votre patience).

J'ai une question pour la parti pg_dump :

Déjà pour être sûr pour les settings :

[...]

Et la (j'ai un doute sur le DB['USER'] mit en commentaire du ticket #16302 de benj) j'ai accès à DATABASES['default']['HOST'] et ['USER'], mais aussi password si je dis pas de bêtise.

Du coup pourquoi passé par la variable d'environnement PGPASSWD?

Parce que pg_dump ne sait pas recevoir de mot de passe sur la ligne de commande, voir man pg_dump:

```
-w
--no-password
    Never issue a password prompt. If the server requires password authentication and a password is not
    available by other means such as a .pgpass file, the
    connection attempt will fail. This option can be useful in batch jobs and scripts where no user is
    present to enter a password.
```

Si on passe par PGPASSWD, est-ce que du coup elle est set au moment de l'appel comme on est connecté avec le tenant?

Pas compris, parle français.

Et ensuite l'appel serait un :

[...]

Et j'ajoute le dump dans le repertoire de backup qui contient aussi le tarball

Oui.

#11 - 22 juillet 2017 11:12 - Jean-Baptiste Jaillet

merci Benj mais du coup j'ai mis à jour la réponse hier soir. (ou j'ai changé le -w et je set bien PGPASSWORD). J'ai mon souci avec les tests.

#12 - 22 juillet 2017 11:27 - Benjamin Dauvergne

Ça ne va pas, il y a déjà une classe de base pour faire des commandes destinées à agir sur un tenant, pas la peine de refaire cela (faut lire du code avant d'en écrire :)) dans hobo.multitenant.management.commands tu as InteractiveTenantOption et TenantWrappedCommand, il vaut mieux partir de InteractiveTenantOption, TenantWrappedCommand servant plutôt à ajouter le support des tenants à une commande existante, comme ceci:

```
from django.db import connection

class MaCommande(InteractiveTenantOption, BaseCommand):

    ....

    def handle(*args, **options):
        tenant = self.get_tenant_from_options_or_interactive(**options)
        connection.set_tenant(tenant)

        <faire ce qu'il y a à faire>
```

Jean-Baptiste Jaillet a écrit :

Bon, j'ai modifié tout ça:

- rajouté le port (connection à distance, le port peut être différent)

Ok, extrait le schéma de connection.tenant.schema_name en utilisant le code plus haut.

Ça je suis sûr que ça ne marche pas:

```
if db_pwd:
    subprocess.check_call(['PGPASSWORD="%s" && export PGPASSWORD' % db_pwd])
```

L'environnement n'est pas maintenue entre plusieurs appels à check_call(), lis un bouquin Unix sur fork/etc.. Le plus simple c'est de faire:

```
import os
os.environ['PGPASSWORD'] = db_pwd
```

Aussi pour formater une date au format ISO, timezone.now().date().isoformat() c'est plus sûr, on finit toujours par se tromper avec les chaînes de formatage.

- le nom de la db (et un truc un peu cheap en cas d'user 'root' qui se connecte et du coup ne trouve pas le schema puisqu'il n'est pas connecté à la base de l'appli. Donc je récupère la base par le nom du service. CF: je demandais conseil plus haut, sûrement un problème de settings des tests multitenant mais j'ai gratté tout ce que je pouvais et j'ai trouvé que ça pour passer les tests. Sinon en temps normal, on récupère le nom de la db (ce qui n'est pas obligatoire) depuis l'entrée DATABASE['default']['NAME']

Incompréhensible, et surtout je ne vois aucune référence à root dans le patch donc je ne sais pas de quoi tu parles, mais oui tout doit venir de DATABASE['default'] sauf le nom du schéma qui vient de connection.tenant.schema_name, c'est tout il n'y a rien d'autre à dire. S'il faut les droits de l'utilisateur du service on les aura car les commande *-manage doivent être lancé avec l'utilisateur du service.

- l'option -w, en relisant bien la doc dans mon patch plus haut j'avais fait ça n'importe comment. Donc je précise -w pour éviter le prompt, et en cas de pwd pour se connecter (ce qui sera le cas chez sictiam puisque connection à distance et chaque db a un pwd), je set PGPASSWORD puis une fois le dump fait j'unset la variable (on ne sait jamais, je me dis que si on la laisse traîner ça peut causer des soucis ensuite).

L'environnement des processus ne fonctionne pas comme ça, il est hérité du processus père et copié dans le processus fils lors d'un fork, ce n'est pas un objet global, il est local au processus donc tes check_call() ne servent à rien.

Du coup les tests passent, mais c'est un peu de la 'triche' :

Les tests passent parce qu'il n'y a pas de mot de passe. Faudra faire des tests à la main.

[...]

Donc encore une fois, je ne sais pas comment sont paramétrés les tests mais depuis le tox il me manque un truc ([...] depuis la racine de l'app django)

C'est un peu tard donc j'espère que ce que j'écris est compréhensible.

Non. :)

Bon d'autres choses;

- quand tu crées un fichier depuis un programme il faut le faire dans un endroit temporaire avec le module [tempfile](#) et si tout se passe bien le déplacer à sa destination. En cas d'erreur il faut nettoyer (supprimer les fichiers temporaires, supprimer récursivement le répertoire de destination qui vient d'être créé, etc...). Le plus simple c'est peut-être de créer un répertoire temporaire de destination avec [tempfile.mkdtemp](#) puis si il y a une exception dans le chemin non-passant (except: quoi) le supprimer avec [shutil.rmtree](#) et dans le chemin passant le déplacer à sa position finale (dans --destination donc) avec [shutil.move](#).
- pour ton tarball, te prends pas la tête avec tarfile il y a toutes les chances que tu te plantes, utilise tar tout bêtement,
- ajoute le nom de l'application dans le nom des fichiers de dump <settings.PROJECT_NAME>__<schema_name>__<date>.sql, on pourra imaginer un jour pouvoir restaurer ce genre de backups simplement depuis les noms,
- étudie la possibilité d'avoir pour destination un zip plutôt qu'un répertoire (et là pareil, on génère d'abord sql et tgz dans des fichiers temporaires et quand tout va bien on ajoute au zip, pour que ce soit transactionnel, qu'on se retrouve pas avec un zip tout seul sans son tarball parce qu'un truc a foiré au milieu)

#13 - 24 juillet 2017 02:29 - Jean-Baptiste Jaillet

- Fichier 0001-multitenant-add-backup_tenant-command-17725.patch ajouté

- Statut changé de Information nécessaire à En cours

OK avec les remarques prises en compte.

Pour la commande, je m'étais inspiré de Delete_tenant puisqu'à la base je devais la modifier et finalement j'en ai fait une nouvelle, et delete_tenant pour le coup n'utilisait pas InteractiveTenantOption. My bad.

Pour le reste j'ai donc fait les fichier dans un dossier temporaire et quand tout est bon, je créer un zip dans le répertoire de destination (en cas d'erreur lors de la création du zip, je supprime le répertoire temporaire et le fichier zip). Je ne fais donc pas de move (quelque soit l'erreur, je flingue le répertoire temporaire).

pour ton tarball, te prends pas la tête avec tarfile il y a toutes les chances que tu te plantes, utilise tar tout bêtement

j'ai cherché j'ai trouvé que tarfile en lib (ou alors j'ai pas compris la remarque, si c'est plus sur le mode d'ouverture w:gz) et je me suis inspiré de ce qui est dans la commande backup de wcs.

Par contre, quand je zip, j'ai bien les deux fichiers mais le tar.gz passe en mode texte et est vide (j'ai gardé mes dossiers temporaires pour voir, le tar.gz est bien construit et contient les bon fichiers) mais quand il est ajouté au zip, je ne sais pas ce qu'il se passe. J'ai lu les trucs sur [zipfile](#) et pas moyen de trouver s'il faut un mode spécifique pour des fichiers archives (sur le net non plus).

Voilà, reste les tests avec une connection distante + pwd (j'ai testé en local, et sur la dev sictiam, et j'ai modifié les tests qui passent aussi en local). Je mets le patch quand même pour l'histoire du tar.gz mal formaté dans le zip et d'éventuelles nouvelles remarques.

#14 - 24 juillet 2017 08:25 - Benjamin Dauvergne

Jean-Baptiste Jaillet a écrit :

pour ton tarball, te prends pas la tête avec tarfile il y a toutes les chances que tu te plantes, utilise tar tout bêtement

j'ai cherché j'ai trouvé que tarfile en lib (ou alors j'ai pas compris la remarque, si c'est plus sur le mode d'ouverture w:gz) et je me suis inspiré de ce qui est dans la commande backup de wcs.

Pour tar je parle de la commande shell, Popen(['tar', 'cvzf', tar_pat, '/var/lib/<app>/<tenant>/'])@.

#15 - 24 juillet 2017 11:34 - Jean-Baptiste Jaillet

- Fichier 0001-multitenant-add-backup_tenant-command-17725.patch ajouté

Ok du coup cette fois le tar.gz est correct dans le zip.

J'ai testé local et sur la recette de sictiam (connexion distante avec authentification) tout passe. Les tests aussi.

#16 - 25 juillet 2017 11:20 - Benjamin Dauvergne

- PROJECT_NAME n'est pas mis dans le nom du fichier zip uniquement dans le nom du dump mais pas dans le nom du tarball, c'est quoi ta logique là ? Tu ne peux pas tout nommer pareil ? Le dump sql l'extension c'est ".sql" pas ".dump" qui n'est pas un format.
- faut chopper toutes les erreurs, sur mkdtemp et sur rmtree notamment (en gros tout ce qui fait des IOs il faut mettre un try/except autour)

On est pas des sauvages:

- "dump failed" -> "backup of postgresql schema %s of database %s failed"
- "targz creation failed" -> "backup of /var/lib/etc./tenant/etc../ failed"

Pour faciliter le traitement des erreurs je te conseillerai de découper ta fonction en fonctions plus petites:

```
def handle(...):
    temp_dir = ...
    try:
        mkdtemp(temp_dir)
    except IOError as e:
        etc..

    try:
        pg_backup_path = self.backup_postgresql(temp_dir)
        tenant_dir_backup_path = self.backup_tenant_dir(temp_dir)
        self.create_zip(temp_dir, pg_backup_path, tenant_dir_backup_path)
    finally:
        try:
            shutil.rmtree(temp_dir)
        except IOError as e:
            print 'cannot remove %s' % temp_dir
```

Ça t'évite d'écrire `shutil.rmtree` partout et on comprend mieux les étapes (je crois qu'on appelle ça de la programmation structurée :)).

#17 - 25 juillet 2017 11:21 - Benjamin Dauvergne

Aussi il faudrait utiliser deux underscores pour séparer application, hostname et date, qu'on puisse facilement faire l'opération inverse.

#18 - 25 juillet 2017 16:06 - Jean-Baptiste Jaillet

- Fichier `0001-multitenant-add-backup_tenant-command-17725.patch` ajouté

Ok, pris en compte.

#19 - 26 juillet 2017 17:42 - Jean-Baptiste Jaillet

- Fichier `0001-multitenant-add-backup_tenant-command-17725.patch` ajouté

Suite aux remarques de [#17726](#) j'ai fait les modifs ici aussi:

- un seul tar du répertoire du tenant auquel on ajoute le dump
- refait les test avec la fixture `tmpdir_factory` de `pytest`
- comme on doit rester cohérent, je préférerais l'idée d'avoir un répertoire backups créé dans le répertoire du tenant en cas de `--destination` non renseigné. Mis à jour dans les tests (avec et sans destination, et vérifier que le répertoire backup n'est pas remis dans l'archive).

Pour les gros points.

#20 - 02 octobre 2017 23:41 - Frédéric Péters

- Assigné à *Jean-Baptiste Jaillet* supprimé

#21 - 07 novembre 2019 16:40 - Benjamin Dauvergne

Termine-t-on ce ticket ? (je veux bien le reprendre si on juge ça utile)

#22 - 07 novembre 2019 17:03 - Thomas Noël

Personnellement ça ne m'a jamais manqué.

#23 - 07 novembre 2019 20:42 - Benjamin Dauvergne

- Statut changé de *En cours* à *Rejeté*

Fichiers

0001-multitenant-add-backup_tenant-command-17725.patch	3,88 ko	21 juillet 2017	Jean-Baptiste Jaillet
0001-multitenant-add-backup_tenant-command-17725.patch	4,52 ko	22 juillet 2017	Jean-Baptiste Jaillet
0001-multitenant-add-backup_tenant-command-17725.patch	5,79 ko	24 juillet 2017	Jean-Baptiste Jaillet
0001-multitenant-add-backup_tenant-command-17725.patch	5,61 ko	24 juillet 2017	Jean-Baptiste Jaillet
0001-multitenant-add-backup_tenant-command-17725.patch	6,65 ko	25 juillet 2017	Jean-Baptiste Jaillet

