Installer - Bug #36238

Installer --reset-data fails on an external DB setup

03/28/2023 07:43 AM - Evgeni Golov

Status:	Closed		
Priority:	Normal		
Assignee:	Evgeni Golov		
Category:	foreman-installer script		
Target version:			
Difficulty:		**Fixed in Releases:**	3.7.0
Triaged:	Yes	**Found in Releases:**	
Bugzilla link:	2053421	**Red Hat JIRA:**	
Pull request:	https://github.com/theforeman/foreman-installer/pull/844		

Description

Cloned from https://bugzilla.redhat.com/show_bug.cgi?id=2053421

Description of problem:
Installer reset on an external DB setup fails, effectively blocking backup restore on that setup.

Version-Release number of selected component (if applicable):
6.10.3 snap 1

How reproducible:
always

Steps to Reproduce:
1. Have an external DB setup (SAT + DB host)
2. On the Satellite, run `satellite-installer --reset-data`

Actual results:
Installer fails:
2022-02-11 04:12:08 [NOTICE] [configure] 1250 configuration steps out of 1850 steps complete.
2022-02-11 04:13:37 [NOTICE] [configure] 1500 configuration steps out of 1850 steps complete.
2022-02-11 04:14:49 [ERROR] [configure] 'pulpcore-manager migrate --noinput' returned 1 instead of one of [0]
2022-02-11 04:14:49 [ERROR] [configure] /Stage[main]/Pulpcore::Database/Pulpcore::Admin[migrate
--noinput]/Exec[pulpcore-manager migrate --noinput]/returns: change from 'notrun' to ['0'] failed: 'pulpcore-manager migrate --noinput'
returned 1 instead of one of [0]
2022-02-11 04:14:49 [NOTICE] [configure] 1750 configuration steps out of 1850 steps complete.
2022-02-11 04:14:52 [NOTICE] [configure] System configuration has finished.

There were errors detected during install.

Looking at the installer log, the real error is:
2022-02-11 03:11:53 [INFO] [configure] /Stage[main]/Pulpcore::Database/Pulpcore::Admin[migrate
--noinput]/Exec[pulpcore-manager migrate --noinput]/returns: Applying rpm.0013_RAW_rpm_evr_extension...Traceback (most
recent call last):
2022-02-11 03:11:53 [INFO] [configure] /Stage[main]/Pulpcore::Database/Pulpcore::Admin[migrate
--noinput]/Exec[pulpcore-manager migrate --noinput]/returns: File "/usr/lib/python3.6/site-packages/django/db/backends/utils.py",
line 82, in _execute
2022-02-11 03:11:53 [INFO] [configure] /Stage[main]/Pulpcore::Database/Pulpcore::Admin[migrate
--noinput]/Exec[pulpcore-manager migrate --noinput]/returns: return self.cursor.execute(sql)
2022-02-11 03:11:53 [INFO] [configure] /Stage[main]/Pulpcore::Database/Pulpcore::Admin[migrate
--noinput]/Exec[pulpcore-manager migrate --noinput]/returns: psycopg2.errors.DuplicateObject: type "pulp_evr_array_item" already
exists

And looking at the migration (
https://github.com/pulp/pulp_rpm/blob/main/pulp_rpm/app/migrations/0013_RAW_rpm_evr_extension.py), it creates types and
functions, not tables. So it's never deleted by
https://github.com/theforeman/foreman-installer/blob/416f45216493f9ad4d331a54d7cc92e42b8dbfd5/hooks/pre/10-reset_data.rb#L6

Expected results:
Successfull reset

Associated revisions

Revision bb67be51 - 03/28/2023 02:55 PM - Evgeni Golov

Fixes #36238 - drop all data owned by the DB user, not only tables

The old logic (introduced all the way back in [1]) took a list of all
tables and executed a `DROP … CASCADE` on them. While this is sufficient
for tables and the **associated** sequences, triggers etc [2][3], it will
not delete objects that have no direct dependency on tables, like types
or functions.

However, we might have DDLs that do create such objects and then after
performing a database reset (thus assuming the DB is empty) the DDL will
fail to re-run with an error like:

```
psycopg2.errors.DuplicateObject: type "pulp_evr_array_item" already exists
```

In a perferct world, those DDLs would be written in a way that they
detect that a certain object already exists and replace it instead.
This is what the problematic pulp DDL [5] does for functions, but
PostgreSQL doesn't support `CREATE OR REPLACE` for types [6].

Instead of trying to obtain a list of all tables and drop those, we can
instruct PostgreSQL to drop all objects owned by a certain user [4].
This has the benefit that it will drop all types of objects, so also
types and functions (but not databases, which is good as we cannot
recreate those without superuser permissions).
The downside is that this will not delete things not owned by the user,
but in our setups I would expect everything to be owned by the
application user as that is the one we use for setup etc and if things
would not be owned by the user, the user would not have been able to
`DROP` it with the old code either.

[1] https://github.com/Katello/katello-installer/pull/550
[2] https://www.postgresql.org/docs/12/sql-droptable.html
[3] https://www.postgresql.org/docs/12/ddl-depend.html
[4] https://www.postgresql.org/docs/12/sql-drop-owned.html
[5]
https://github.com/pulp/pulp_rpm/blob/06ed3c2b214684b06186d32b6a7c4ff018efa2c1/pulp_rpm/app/migrations/0013_RAW_rpm_evr_extension.py
[6] https://www.postgresql.org/docs/12/sql-createtype.html

History

#1 - 03/28/2023 07:44 AM - Evgeni Golov

*- Subject changed from Installer --reset-data fails on an external DB setup
to Installer --reset-data fails on an external DB setup*

- Category set to foreman-installer script

#2 - 03/28/2023 08:03 AM - The Foreman Bot

- Status changed from New to Ready For Testing

- Pull request https://github.com/theforeman/foreman-installer/pull/844 added

#3 - 03/28/2023 02:55 PM - The Foreman Bot

- Fixed in Releases 3.7.0 added

#4 - 03/28/2023 03:01 PM - Evgeni Golov

- Status changed from Ready For Testing to Closed

Applied in changeset [installer|bb67be51bab0ea4d7e3498b03c549dd99c731598](#).

#5 - 05/25/2023 11:49 AM - Ewoud Kohl van Wijngaarden

- Triaged changed from No to Yes