

RhodeCode CE/EE - Support #5503

failed to upgrade to 4.13.3

25.10.2018 21:32 - Stephen Serafin

Status:	New	Start date:	25.10.2018
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Sorting:		Affected Version:	v4.13
Commit Number:			
Description			
Upgrading from 4.11.6 to 4.13.3 rccontrol from 1.17.1 to 1.19.2 vcserver-1 from 4.11.6 to 4.13.3 works enterprise-1 from 4.11.6 to 4.13.3 fails. Database updates ATTEMPTING TO UPGRADE DATABASE FROM VERSION 85 TO VERSION 90 upgrade step 86 to 89 run followed by a sqla base reset sqlalchemy.exc.ProgrammingError: (psycopg2.ProgrammingError) there is no unique constraint matching given keys for referenced table "repositories" [SQL: "\nCREATE TABLE user_to_repo_branch_permissions (\n\tbranch_rule_id SERIAL NOT NULL, \n\trepository_id INTEGER NOT NULL, \n\tpermission_id INTEGER NOT NULL, \n\trole_to_perm_id INTEGER NOT NULL, \n\trole_order INTEGER NOT NULL, \n\tbranch_pattern TEXT, \n\tbranch_hash TEXT, \n\tPRIMARY KEY (branch_rule_id), \n\tFOREIGN KEY(repository_id) REFERENCES repositories (repo_id), \n\tFOREIGN KEY(permission_id) REFERENCES permissions (permission_id), \n\tFOREIGN KEY(role_to_perm_id) REFERENCES repo_to_perm (repo_to_perm_id)\n)\n\n"] Upgrade of RhodeCode Enterprise failed. Rolling back to previous version. Attached is the output from the upgrade session			

History

#1 - 25.10.2018 23:15 - Daniel D

That is odd and unexpected. I'm assuming repositories table is in a bad state somehow.

Can you check on your postgres if repositories (repo_id) column is unique?

Was there any upgrade problems in older days related to database?

Best,

#2 - 30.10.2018 00:53 - Daniel D

Somehow this got into SPAM:

Postgres has unique repo_id for each row but they are not sequential:

```
rhodecode=> SELECT repo_id FROM repositories;
```

```
repo_id
```

```
-----
```

```
1
```

```
3
```

6
7
9
10
11
12
13
14
15
19
20
21
22
23
24
26
27
28
30
32
33
34
35
36
37
38
39
29

(30 rows)

mysql to postgres migration steps:

```
# yum install pgloader-3.4.1-3.rhel7.x86_64.rpm
```

```
# su - rhodecode
```

```
$ cat<<EOF>>migration_command
```

```
load database
```

```
from mysql://rhodecode:see-my.cnf-for-this-password@localhost:3306/rhodecode
```

```
into postgresql://rhodecode:see-pgpass-for-this-password@127.0.0.1/rhodecode
```

```
WITH truncate,
```

```
create no tables,
```

```
create no indexes,
```

```
reset no sequences
```

```
ALTER schema 'rhodecode' rename to 'public'
```

```
CAST type datetime to timestamp
```

```
;
```

```
EOF
```

```
$ ssh -L 3306:127.0.0.1:3306 username_mgr@mysqlhostname -N -f
```

```
$ scl enable rh-postgresql96 -- pgloader --dry-run migration_command
```

```
## make sure connection to both databases was successful
```

```
$ scl enable rh-postgresql96 -- pgloader migration_command
```

```

$ mv /tmp/pgloader/pgloader.log .

## make sure the data looks reasonable and the license_key made it

$ scl enable rh-postgresql96 -- psql rhodecode

rhodecode=> SELECT app_settings_value FROM rhodecode_settings WHERE app_settings_name = 'license_key';

## Fixup the database sequencing

$ scl enable rh-postgresql96 -- psql rhodecode

DO

$$

DECLARE

    seqname text;

    nspname text;

    seqval bigint;

BEGIN

    FOR nspname, seqname IN select n.nspname, c.relname FROM

pg_class c INNER JOIN pg_namespace n ON (c.relnamespace = n.oid)

WHERE c.relkind = 'S' AND n.nspname = 'public' LOOP

        EXECUTE format('SELECT last_value FROM %I.%I', nspname, seqname) INTO seqval;

        PERFORM setval(quote_ident(nspname)||'.'||quote_ident(seqname), seqval + 10000);

    END LOOP;

END;

$$ LANGUAGE plpgsql;

\q

```

#3 - 30.10.2018 00:55 - Daniel D

Can you check please if repo_id has an UNIQUE index set for Postgres? This is a requirement for this to continue, maybe the migration didn't convert that properly?

#4 - 30.10.2018 15:05 - Stephen Serafin

Is this what you are looking for:

```

rhodecode=> \d repositories_repo_id_seq
Sequence "public.repositories_repo_id_seq"
Column | Type | Value
-----+-----+-----
sequence_name | name | repositories_repo_id_seq
last_value | bigint | 10001
start_value | bigint | 1
increment_by | bigint | 1
max_value | bigint | 9223372036854775807
min_value | bigint | 1
cache_value | bigint | 1
log_cnt | bigint | 0
is_cycled | boolean | f
is_called | boolean | t
Owned by: public.repositories.repo_id

```

#5 - 30.10.2018 15:07 - Daniel D

Can you show output as in this example:

```

rhodecode=# select * from pg_indexes where tablename = 'repositories';
 schemaname | tablename | indexname | tablespace |
-----+-----+-----+-----+
                indexdef
-----+-----+-----+-----+
 public    | repositories | repositories_pkey |          | CREATE UNIQUE INDEX repositories_pkey ON public.repositories USING btree (repo_id)
 public    | repositories | repositories_repo_name_hash_key |          | CREATE UNIQUE INDEX repositories_repo_name_hash_key ON public.repositories USING btree (repo_name_hash)
 public    | repositories | r_repo_name_idx |          | CREATE INDEX r_repo_name_idx ON public.repositories USING btree (repo_name)
(3 rows)

```

If you're missing any of those, please create as in indexdef example.

#6 - 30.10.2018 15:19 - Stephen Serafin

- File *pg_indexes.gz* added

The migration includes a "create no indexes" as per the sqlite to postgres example on the wiki.
Are there any other indexes to consider?

Included is the output from
select * from pg_indexes;

#7 - 30.10.2018 16:29 - Stephen Serafin

- File *rhocode_migration.gz* added

Seeing some of the indexes on the source database it seems the migration plan needs modification.

Here is the goal:

From RHEL6 to RHEL7

From mysql 5.1.73 to postgres 9.6.10

From rabbitmq 3.1.5 to rabbitmq 3.7.8

From elasticsearch 2.3.0 to elasticsearch-oss 6.3.1

included are the steps for the rhocode pieces

#8 - 06.11.2018 11:57 - Daniel D

Stephen -

The create no indexes refers to the old database that is a source of data.

I believe that if you have a NEW database created it comes with all indexes and requirements meet. Then Pgloader just transfers the data.
Would you have any idea why the creation of initial database would lost indexes?

#9 - 06.11.2018 18:28 - Daniel D

I wonder if maybe we should try to do a PG -> PG migration to get the indexes back in place?

Files

failedupgrade.gz	6.08 KB	25.10.2018	Stephen Serafin
pg_indexes.gz	2.56 KB	30.10.2018	Stephen Serafin
rhocode_migration.gz	2.14 KB	30.10.2018	Stephen Serafin