

Foreman - Bug #32832

Host auto provisioning allows an existing host to be inserted again

06/17/2021 10:38 PM - Jean Lubatti

Status: New	
Priority: Normal	
Assignee:	
Category: Host registration	
Target version:	
Difficulty: easy	Fixed in Releases:
Triaged: No	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request:	
Description	
Hello,	
We are operating a foreman instance with a few thousand hosts and relying on puppet/foreman for automation.	
We observe a rare and seemingly extremely random behavior where puppet masters end up (once every few months or so) pushing for a single host strange facts to foreman. That triggers a "re-subscribe" of the host with mostly the same characteristics.	
We are aware of the "Create new hosts when facts are uploaded" option and we have set it to "no" following this investigation.	
However, we believe that in any case, allowing an existing host to be "re-entered" in the database on top of an existing record creates a very dangerous and unpredictable behavior especially considering that foreman is a pivot for the automation stack (values fed for puppet for example will depend on which entry will be returned by queries on the hosts table out of the two present values).	
As such, I would propose either:	
1/ enforcing unicity on the column name (that contains the FQDN) in the host table in order to maintain consistency no matter what (dynamic recording or not) directly in the database as such:	
<pre>CREATE UNIQUE INDEX host_name_unique_idx ON hosts(name);</pre>	
2/ alternatively ensure that the automatic registration/creation of hosts is validating unicity even when automatically called on new facts or reports creation.	
Best Regards,	
-Jean	
Related issues:	
Related to Foreman - Feature #32857: Create unique database constraint on hos...	New

History

#1 - 06/18/2021 06:23 AM - Lukas Zapletal

This comes from some RoR limitation, let me bring this to the broader audience:

<https://community.theforeman.org/t/why-dont-we-have-unique-index-on-host-names/24071>

We do not use SQL transactions for some key operations (mostly host edits) because it does not work with our orchestration system which is based on ActiveRecord callbacks. That does not help much too and these inconsistencies are unfortunately common.

#2 - 06/18/2021 08:20 AM - Ewoud Kohl van Wijngaarden

For what it's worth, especially with Puppet we should rely on certname. That should have a unique constraint. If it doesn't, that's probably a bug.

#3 - 06/18/2021 11:25 AM - Jean Lubatti

In case it helps, we have noted that in the hosts table, the conflicting data record is always (obviously) the ones with a greater auto-increment ID:

```
foreman=# select id,name from hosts where name = 'XXXXXXXX.XXXXXX.XX.XX.XXXXXX.XXX';
id | name |
4211 | XXXXXXXX.XXXXXX.XX.XX.XXXXXX.XXX |
3046 | XXXXXXXX.XXXXXX.XX.XX.XXXXXX.XXX |
(2 rows)
```

It is a bit crude, but we deleted the records directly prior to applying the index like this:

```
foreman=# BEGIN;
BEGIN
foreman=# DELETE from fact_values where host_id = 4211;
DELETE 829
foreman=# DELETE from host_facets_reported_data_facets where host_id = 4211;
DELETE 1
foreman=# DELETE from host_classes where host_id = 4211;
DELETE 0
foreman=# DELETE from host_status where host_id = 4211;
DELETE 1
foreman=# DELETE from nics where host_id = 4211;
DELETE 13
foreman=# DELETE from reports where host_id = 4211;
DELETE 1
foreman=# DELETE from tokens where host_id = 4211;
DELETE 0
foreman=# DELETE from hosts where id=4211;
DELETE 1
foreman=# COMMIT;
```

Out of several thousand hosts, we only had 3 inconsistencies accros organizations and 1 inconsistency within the same organization. In all cases, the correct entry was always following logic (e.g: the oldest entry was the correct one).

Best Regards,

-Jean

#4 - 06/23/2021 08:59 AM - Lukas Zapletal

- Related to Feature #32857: Create unique database constraint on host name added

#5 - 06/23/2021 09:01 AM - Lukas Zapletal

Thanks again for the repost and elaborated details. I am moving this to [#32857](#)