Description

In #18618, the initialization code was moved from foreman-tasks to foreman. There was however one thing done differently, where the initialization of dynflow world was moved directly to the time `Rails.application.dynflow` was called. This is not right, as there are situations where plugins go and call `Rails.application.dynflow.config` before the whole stack is prepared, which leads to database connection being opened too early, which can cause the whole lot of issues (as we've hit currently with Katello).

After we released #18673, the Katello tests started failing on

```
ActiveRecord::StatementInvalid: PG::ObjectInUse: ERROR:  database "test_develop_pr_katello-2-test" is being accessed by other users
DETAIL:  There are 3 other sessions using the database.
```

That was due to fact that the dynflow world was initialized too early.

Related issues:

- Related to Foreman - Feature #18618: Use Dynflow as an ActiveJob backend Closed 02/22/2017
- Related to foreman-tasks - Bug #18673: Remove initialization code in favor of... Closed 02/24/2017

Associated revisions

Revision 0f225c99 - 09/13/2017 04:56 AM - Ivan Necas

Fixes #20908 - postpone initialization of Dynflow runtime

In #18618, the initialization code was moved from foreman-tasks to foreman. There was however one thing done differently, where the initialization of dynflow world was moved directly to the time `Rails.application.dynflow` was called. This is not right, as there are situations where plugins go and call `Rails.application.dynflow.config` before the whole stack is prepared, which leads to database connection being opened too early, which can cause the whole lot of issues (as we've hit currently with Katello).

There was also change in the signature of `Dynflow::Rails#initialize` method, that needs to be updated.
In #18618, the initialization code was moved from foreman-tasks to foreman. There was however one thing done differently, where the initialization of dynflow world was moved directly to the time `Rails.application.dynflow` was called. This is not right, as there are situations where plugins go and call `Rails.application.dynflow.config` before the whole stack is prepared, which leads to database connection being opened too early, which can cause the whole lot of issues (as we've hit currently with Katello).

There was also change in the signature of `Dynflow::Rails#initialize` method, that needs to be updated.

---

Historical Notes

**#1 - 09/12/2017 03:32 PM - Ivan Necas**
- Legacy Backlogs Release (now unused) set to 240

We should handle this before 1.16 gets out

**#2 - 09/12/2017 03:32 PM - Ivan Necas**
- Related to Feature #18618: Use Dynflow as an ActiveJob backend added

**#3 - 09/12/2017 03:33 PM - Ivan Necas**
- Related to Bug #18673: Remove initialization code in favor of Dynflow added

**#4 - 09/12/2017 03:39 PM - The Foreman Bot**
- Status changed from Assigned to Ready For Testing
- Pull request https://github.com/theforeman/foreman/pull/4826 added

**#5 - 09/13/2017 05:01 AM - Ivan Necas**
- Status changed from Ready For Testing to Closed
- % Done changed from 0 to 100

Applied in changeset 0f225c99973cf3e67b21550865e7dc990eecd6ff.