

Pulp - Story #2965

As a user, I can use pulp_example plugin to import content

08/08/2017 09:11 PM - dkliban@redhat.com

Status:	MODIFIED	Start date:	
Priority:	High	Due date:	
Assignee:	dkliban@redhat.com	% Done:	0%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:	3.0	QA Contact:	
Platform Release:		Complexity:	
Blocks Release:		Smash Test:	
Backwards Incompatible:	No	Verified:	No
Groomed:	Yes	Verification Required:	No
Sprint Candidate:	Yes	Sprint:	Sprint 26
Tags:			

Description

The Example plugin will provide a setup.py for installing the plugin. The install will create an entry point that will make the plugin discover-able by Pulp.

The Example plugin will provide ExampleContent content type to Pulp. ExampleContent will be exactly the same as FileContent from file plugin.

The Example plugin will also provide an ExampleImporter. The importer will

```
- perform downloads asynchronously and in parallel
- minimize number of select queries executed to look up existing content and artifacts
- use separate database transactions for each content unit created to avoid having to perform database cleanup when failures occur
```

The pulp_example repository will also include a README that will provide documentation about the plugin.

History

#1 - 08/08/2017 09:29 PM - mhrivnak

Here you go! https://github.com/pulp/pulp_example

#2 - 08/08/2017 09:35 PM - bmbouter

Will this also include a setup.py?

Will this also include an entrypoint to be installed with?

Also a basic README in the repo would be nice.

#3 - 08/08/2017 09:48 PM - dkliban@redhat.com

- Description updated

#4 - 08/08/2017 09:48 PM - dkliban@redhat.com

Yes and yes. I updated the description.

#5 - 08/08/2017 10:03 PM - jortel@redhat.com

When using transactions, it's very important to insert in the same order to prevent deadlock (table locks).

The ChangeSet inserts in this order:

- content (base)
- content (concrete)
- artifact
- content-artifact
- deferred-artifact

Suggest this plugin do the same. We can discuss the order itself (a little) but Pulp must require plugins (using transactions, and they all should) to insert in a known order.

#6 - 08/08/2017 10:12 PM - bmbouter

I think Django will fail to save if the order isn't correct because the remote objects won't have primary keys set yet. If that is true then during development this should be easy to get correct because it work otherwise.

Can you describe this table lock in more detail? An example would also be great.

#7 - 08/10/2017 03:06 PM - dkliban@redhat.com

- Status changed from NEW to ASSIGNED

- Assignee set to dkliban@redhat.com

#8 - 08/10/2017 03:36 PM - jortel@redhat.com

- Groomed changed from No to Yes

#9 - 08/10/2017 04:12 PM - mhrivnak

- Sprint/Milestone set to 43

#10 - 08/10/2017 04:32 PM - mhrivnak

- Priority changed from Normal to High

#11 - 08/10/2017 04:45 PM - jortel@redhat.com

bmbouter wrote:

I think Django will fail to save if the order isn't correct because the remote objects won't have primary keys set yet. If that is true then during development this should be easy to get correct because it work otherwise.

Can you describe this table lock in more detail? An example would also be great.

Right, the DB will enforce referential integrity based on constraint created by django. I'm talking about something else. While in a transaction, if an insert is attempted, the RDBMS will acquire a table lock and insert the row but not commit it. That lock will be held for the duration of the transaction. Inserts are performed on 2+ tables in opposite order in different transactions results in deadlock.

Much of the order I mentioned in comment 5 is dictated by DB constraints. However, Content and Artifact can be inserted in any order.

Example:

T = transaction

```
T-1: insert into Content
Content table lock held by T-1

T-2: insert into Artifact
Artifact table lock held by T-2

T-1: insert into Artifact
Wait on Artifact lock held by T-2

T-2: insert into Content
Wait on Content lock held by T-1
```

This is a classic RDBMS problem and the traditional solution is to ensure that operations (like insert) that results in table locks are performed in a consistent order. PostgreSQL may have some algorithm to sort this out but I doubt it. I can run a quick test to substantiate this issue if needed.

#12 - 09/05/2017 06:04 PM - jortel@redhat.com

- Sprint/Milestone changed from 43 to 44

#13 - 09/25/2017 03:09 PM - mhrivnak

- Sprint/Milestone changed from 44 to 45

#14 - 09/25/2017 05:11 PM - dkliban@redhat.com

- Status changed from ASSIGNED to MODIFIED

https://github.com/pulp/pulp_example/pull/1

#15 - 12/19/2017 04:24 PM - bmbouter

- Tags deleted (Pulp 3 Plugin Writer Alpha)

Cleaning up Redmine tags

#16 - 03/09/2018 12:24 AM - bmbouter

- Sprint set to Sprint 26

#17 - 03/09/2018 12:24 AM - bmbouter

- Sprint/Milestone deleted (45)

#18 - 04/25/2019 06:46 PM - davidddavis

- Sprint/Milestone set to 3.0

#19 - 04/26/2019 10:38 PM - bmbouter

- Tags deleted (Pulp 3)