

## Pulp - Issue #6045

### Pulp content app loses database connection

01/23/2020 03:11 PM - osapryki

<b>Status:</b>	CLOSED - CURRENTRELEASE	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	daviddavis	<b>Estimated time:</b>	0:00 hour
<b>Category:</b>		<b>Groomed:</b>	No
<b>Sprint/Milestone:</b>	3.1.1	<b>Sprint Candidate:</b>	No
<b>Severity:</b>	2. Medium	<b>Tags:</b>	
<b>Version:</b>		<b>Sprint:</b>	Sprint 66
<b>Platform Release:</b>		<b>Quarter:</b>	
<b>OS:</b>			
<b>Triaged:</b>	Yes		

#### Description

After a long time running pulp-content-app all requests begin failing with the following error message:

Traceback (most recent call last):

```
File "/venv/lib64/python3.6/site-packages/aiohttp/web_protocol.py", line 418, in start
    resp = await task
File "/venv/lib64/python3.6/site-packages/aiohttp/web_app.py", line 458, in _handle
    resp = await handler(request)
File "/venv/lib64/python3.6/site-packages/pulpcore/content/handler.py", line 117, in stream_content
    return await self._match_and_stream(path, request)
File "/venv/lib64/python3.6/site-packages/pulpcore/content/handler.py", line 309, in _match_and_stream
    distro = self._match_distribution(path)
File "/venv/lib64/python3.6/site-packages/pulpcore/content/handler.py", line 158, in _match_distribution
    return BaseDistribution.objects.get(base_path__in=base_paths).cast()
File "/venv/lib64/python3.6/site-packages/django/db/models/manager.py", line 82, in manager_method
    return getattr(self.get_queryset(), name)(*args, **kwargs)
File "/venv/lib64/python3.6/site-packages/django/db/models/query.py", line 402, in get
    num = len(clone)
File "/venv/lib64/python3.6/site-packages/django/db/models/query.py", line 256, in __len__
    self._fetch_all()
File "/venv/lib64/python3.6/site-packages/django/db/models/query.py", line 1242, in _fetch_all
    self._result_cache = list(self._iterable_class(self))
File "/venv/lib64/python3.6/site-packages/django/db/models/query.py", line 55, in __iter__
    results = compiler.execute_sql(chunked_fetch=self.chunked_fetch, chunk_size=self.chunk_size)
File "/venv/lib64/python3.6/site-packages/django/db/models/sql/compiler.py", line 1131, in execute_sql
    cursor = self.connection.cursor()
File "/venv/lib64/python3.6/site-packages/django/db/backends/base/base.py", line 256, in cursor
    return self._cursor()
File "/venv/lib64/python3.6/site-packages/django/db/backends/base/base.py", line 235, in _cursor
    return self._prepare_cursor(self.create_cursor(name))
File "/venv/lib64/python3.6/site-packages/django/db/utils.py", line 89, in __exit__
    raise dj_exc_value.with_traceback(traceback) from exc_value
File "/venv/lib64/python3.6/site-packages/django/db/backends/base/base.py", line 235, in _cursor
    return self._prepare_cursor(self.create_cursor(name))
File "/venv/lib64/python3.6/site-packages/django/db/backends/postgresql/base.py", line 223, in create_cursor
    cursor = self.connection.cursor()
django.db.utils.InterfaceError: connection already closed
```

Possible reason is that pulp-content-app doesn't re-establish database connection whenever it closes.

**Related issues:**

Related to Pulp - Issue #9276: Content app can have unusable/closed db connec...

**MODIFIED****Associated revisions****Revision 5aabb202 - 02/11/2020 09:36 PM - daviddavis**

Fix "connection already closed" error in content app

fixes #6045 <https://pulp.plan.io/issues/6045>**Revision 6fe74ee2 - 02/17/2020 06:29 PM - bmbouter**

Add bugfix changelog entry for 6045

<https://pulp.plan.io/issues/6045> re #6045**Revision 3da5c467 - 02/17/2020 09:27 PM - daviddavis**

Fix "connection already closed" error in content app

fixes #6045 <https://pulp.plan.io/issues/6045>

(cherry picked from commit 5aabb202dd59cbe2d30ef5ad91f01932c9ca041b)

**Revision 3293134e - 02/17/2020 09:27 PM - bmbouter**

Add bugfix changelog entry for 6045

<https://pulp.plan.io/issues/6045> re #6045

(cherry picked from commit 6fe74ee2978732c72dcfd8a8a6eed5760a4dee7)

**History****#1 - 01/23/2020 05:50 PM - bmbouter**Thank you for reporting this. I expect Django to re-establish the connection. I see this from its management docs [here](#)

How can I reproduce this issue also, any pointers for me?

**#2 - 01/28/2020 05:27 PM - osapryki**

I think this discussion might be related:

<https://stackoverflow.com/questions/31504591/interfaceerror-connection-already-closed-using-django-celery-scrapy>**#3 - 01/28/2020 11:34 PM - bmbouter**

This error suggests the connection is closing from the Postgresql side. I was chatting about the issue in their channel and they indicated Postgresql is not closing the connection. They suggested that a firewall in the middle (openshift perhaps?) is firewalling the idle connection.

Was there anything in the postgresql logs that indicates it is closing the connection or that django maybe closed it (aka the firewall?)

**#4 - 01/29/2020 02:57 PM - bmbouter**

I had two ideas.

One: we could check the postgresql connection during the pre\_request event <http://docs.gunicorn.org/en/latest/settings.html#pre-request> but it would be costly in terms of the request-response runtime increase.

Two: I looked for some sort of process recycling in gunicorn (which runs the content app) but I didn't see it.

**#5 - 02/10/2020 08:34 PM - ironfroggy**[bmbouter](#), maybe you want gunicorn's max\_requests setting, which will restart workers after N number of requests handled?<https://docs.gunicorn.org/en/stable/settings.html#max-requests>**#6 - 02/11/2020 07:32 PM - osapryki***- Description updated*

@ironfroggy Limiting number of connections won't help because connection can terminate any time within this limit.

RCA: Django manages connections implicitly. It sets up signal handlers to handle dead or expired connections before and after each request [1]. This

handler executes `close_if_unusable_or_obsolete` method, which closes connection if it times out when `CONN_MAX_AGE` is exceeded or if `is_usable()` [3] check fails.

[1] [https://github.com/django/django/blob/stable/2.2.x/django/db/\\_init\\_.py#L60](https://github.com/django/django/blob/stable/2.2.x/django/db/_init_.py#L60)

[2] <https://github.com/django/django/blob/stable/2.2.x/django/db/backends/base/base.py#L492>

[3] <https://github.com/django/django/blob/stable/2.2.x/django/db/backends/postgresql/base.py#L249>

Since content-app is asyncio application and uses django connection, it will share single connection between coroutines. Possible solution is to close the connection when request handler returns response or after significant part of database queries.

```
from django.db import connection

# Put this either after significant database communication logic or at the beginning or at the end of request handler.
connection.close()
```

Also since the connection is shared between coroutines you should be extremely careful with that to avoid connection state being inconsistent between coroutine context switches. For example you should make sure **context is never switched within a transaction**.

**#7 - 02/11/2020 08:02 PM - daviddavis**

- Triaged changed from No to Yes
- Sprint set to Sprint 66

**#8 - 02/11/2020 09:00 PM - daviddavis**

- Status changed from NEW to POST

<https://github.com/pulp/pulpcore/pull/535>

**#9 - 02/11/2020 09:57 PM - daviddavis**

- Status changed from POST to MODIFIED

Applied in changeset [pulpcore|5aabb202dd59cbe2d30ef5ad91f01932c9ca041b.](https://pulpcore.org/commits/5aabb202dd59cbe2d30ef5ad91f01932c9ca041b/)

**#10 - 02/17/2020 02:47 PM - daviddavis**

- Assignee set to daviddavis

**#11 - 02/17/2020 09:27 PM - daviddavis**

Applied in changeset [pulpcore|3da5c4672a6b047e5eee9b58d6a9400676ea26b9.](https://pulpcore.org/commits/3da5c4672a6b047e5eee9b58d6a9400676ea26b9/)

**#12 - 02/18/2020 04:12 PM - bmbouter**

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE

**#13 - 02/18/2020 04:13 PM - bmbouter**

- Sprint/Milestone set to 3.1.1

**#14 - 08/23/2021 05:38 PM - ttereshc**

- Related to Issue #9276: Content app can have unusable/closed db connections in pulpcore 3.15/3.16 added