

Python GMN - Bug #7240

Critical bug in GMN process_replication_queue

2015-06-29 18:11 - Dave Vieglais

Status: Closed	Start date: 2015-06-29
Priority: Urgent	Due date:
Assignee: Mark Flynn	% Done: 100%
Category:	Estimated time: 0.00 hour
Target version:	
Story Points:	
Description	
process_replication_queue uses a file lock to ensure only one instance is running at a time.	
However, the implementation is incorrect and does not provide the expected functionality. The method:	
<pre>def abort_if_other_instance_is_running(self): single_path = os.path.join(tempfile.gettempdir(), os.path.splitext(file_)[0] + '.single') f = open(single_path, 'w') try: fcntl.lockf(f, fcntl.LOCK_EX fcntl.LOCK_NB) except IOError: logging.info('Aborted: Another instance is still running') exit(0)</pre>	
if used to create and test for the lock, however the file descriptor is released as soon as the method exits, which removes the lock, and subsequently allows other instances to start up.	
The file descriptor should be held for the life of the application or alternatively, simply create a file, then treat existence of the file as the existence of a lock.	
Related issues:	
Related to Python GMN - Bug #7295: Critical bug in GMN process_system_metadat...	Closed 2015-08-17

History

#1 - 2015-06-29 20:26 - Roger Dahl

Dave Vieglais wrote:

However, the implementation is incorrect and does not provide the expected functionality.

Oops...

The file descriptor should be held for the life of the application or alternatively, simply create a file, then treat existence of the file as the existence of a lock.

Adding global f at the top of the function should do the trick.

Just treating the existence of the file as a lock can be problematic if something happens that causes the script to bomb out without deleting the file, e.g., if it gets killed by the OOM.

#2 - 2015-06-29 21:33 - Dave Vieglais

Can't imagine what could go wrong declaring "f" as a global...

Better to declare the variable within the class, and give it a real name.

See <http://blog.vmfarms.com/2011/03/cross-process-locking-and.html> for a simple example that provides a reusable Lock class

#3 - 2015-07-06 21:38 - Mark Flynn

I added the Lock class from the link Dave included and used that to lock the resource while the queue is processed::

```
def abort_if_other_instance_is_running(self):
    single_path = os.path.join(tempfile.gettempdir(),
                               os.path.splitext(file_)[0] + '.single')
    try:
        lock = Lock(single_path)
        lock.acquire()
        #fcntl.lockf(f,fcntl.LOCK_EX |fcntl.LOCK_NB)
    except IOError:
        logging.info('Aborted: Another instance is still running')
        exit(0)
    return lock

def handle_noargs(self, **options):
    try:
        verbosity = int(options.get('verbosity', 1))
        self._log_setup(verbosity)
        logging.debug('Running management command: process_replication_queue')
        lock = self._abort_if_other_instance_is_running()
        self._abort_if_stand_alone_instance()
        p = ReplicationQueueProcessor()
        p.process_replication_queue()
    finally:
        lock.release()
```

I'm trying write to functional tests, but it requires makes calls using cnclient to a CN. I can verify that the lock works using the debugger. When I pause execution after the lock has been acquired and run another instance of process_replication_queue, the second instance pauses until the first is done.

#4 - 2015-07-11 01:27 - Mark Flynn

I have written unit tests that test the basic functionality of the file locks and those work. However, I'm not sure how to functionally test `process_replication_queue`. I don't see where an instance of class `Command` is created so that `_abort_if_other_instance_is_running()` can lock the `.single` file

#5 - 2015-07-16 16:57 - Mark Flynn

I see that 'Command' is a django requirement for writing custom django commands.

#6 - 2015-07-16 21:49 - Mark Flynn

I've finished fixing the bug and posted the code and tests to subversion. I'll work with Roger to get it tagged and released to pypi.

#7 - 2015-08-17 21:59 - Mark Servilla

- Related to Bug #7295: Critical bug in GMN `process_system_metadata_refresh_queue` added

#8 - 2015-10-06 18:30 - Mark Flynn

- Status changed from New to Closed

- % Done changed from 0 to 100

fixed bug, created unit tests to ensure file is locked and ran functional test to show that attempts to create run `process_replication_queue` fail if it is already running