

w.c.s. - Development #56009

cron : récupérer les erreurs dans chaque cron_worker

06 août 2021 01:29 - Benjamin Dauvergne

Statut:	Fermé	Début:	06 août 2021
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Actuellement une erreur en dehors de job.function va arrêter complètement le déroulement des crons, il faudrait au minimum encadrer d'un try/except l'exécution pour chaque tenant. Il me semble qu'avant il y avait un fork entre les deux qui apportait cette protection.			

Historique

#2 - 06 août 2021 08:28 - Frédéric Péters

il y avait un fork

Avant 2014 ça ne passait pas par cron mais par un processus interne qui faisait en effet un fork par tenant. ([#6735](#) introduit l'exécution via cron et retire ça).

..

Mais les erreurs qui ne sont pas attrapées on les voit normalement sur stdout/stderr et cron nous les envoie par email, ce qui nous permet de réagir, il y a un truc qui échoue ici ?

#3 - 06 août 2021 09:10 - Benjamin Dauvergne

Frédéric Péters a écrit :

Mais les erreurs qui ne sont pas attrapées on les voit normalement sur stdout/stderr et cron nous les envoie par email, ce qui nous permet de réagir, il y a un truc qui échoue ici ?

Oui, le fait que /etc/aliases contienne juste root: sictiam, je vais ajouter admin+sictiam-prod@ à aliases, c'était fait sur sictiam-test mais pas sictiam-prod.

Mais là l'intérêt que j'y vois c'est qu'avec une isolation par tenant on aurait eu un bug que sur un seul tenant, ici on a eu un bug sur tous les tenants qui suivaient dans la boucle.

#5 - 06 août 2021 09:22 - Frédéric Péters

- Fichier 0001-cron-do-not-abort-on-unhandled-exceptions-by-default.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Quelque chose de cet ordre ?

#6 - 06 août 2021 09:37 - Benjamin Dauvergne

Oui mais il me semble que dans la situation précédente à défaut d'avoir les erreurs circonscrites à un tenant, on devait avoir la trace python sur stdout/stderr comme tu le dis (et sur sentry qui pose aussi un handler global d'exception, c'est d'ailleurs grâce à ça que j'ai pu voir le problème).

Là on va perdre les deux et on aura que le nom de l'exception et du tenant, je ferai donc un `bête logging.exception(...)` en espérant qu'un StreamHandler tout bête soit présent sur le root logger (je ne sais pas actuellement).

#7 - 06 août 2021 09:52 - Frédéric Péters

logging.exception

On ne fait ça nulle part dans w.c.s. aucune idée de comment ça serait supposé marcher le packaging fait

```
# don't rely on hobo logging as it requires hobo multitenant support.  
LOGGING = {}
```

et ça m'évite de penser à ce module que je ne comprends pas. Et malgré ça avec le passage à 2.2 il y a des trucs qui ont changé et des trucs loggués en double et je ne sais pas.

Je ne vais rien proposer qui dépendrait de logging.

#8 - 06 août 2021 10:42 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

C'était une suggestion de forme,

```
from wcs.qommon.publisher import get_logger  
...  
get_logger().exception(...)
```

me va tout aussi bien, et c'est ce qui est fait ailleurs; reste que je ne crois pas que ça va afficher quelque chose sur stderr/stdout, ça va tout balancer (mais mal) dans tenants/<host>/wcs.log et heureusement aussi dans sentry qui a un hook global sur la machinerie de logging.

Il me semble que get_publisher().record_error affiche sur la sortie standard via le Logger maison de quixote, mais maintenant c'est uniquement sur la première occurrence (Je suppose que c'est pareil pour le mail avec trace). Si ça intéresse quelqu'un je trouverai utile qu'on y mette un peu d'ordre parce qu'on a 2/3 chemins de log actuellement qui se comportent tous un peu différemment (get_logger(), record_error, qui envoie des mails et crée des objets LoggedError).

Ça me va qu'on vire get_logger() et qu'on standardise sur record_error mais il faudrait que ça gère sentry aussi (on voit bien dans sentry qu'on a aucune des erreurs journalisée par record_error(), certainement qu'on ne veut pas les erreurs de conditions/expression/gabarits mais on veut les autres) que certains utilisent.

#9 - 06 août 2021 10:42 - Benjamin Dauvergne

- Assigné à Benjamin Dauvergne supprimé

Oups.

#10 - 15 janvier 2024 20:44 - Frédéric Péters

- Statut changé de Solution proposée à Fermé

La procédure d'exécution des cron a été totalement revue ([#33280](#)) et il y a du code pour gérer sentry ([#78668](#)).

Fichiers

0001-cron-do-not-abort-on-unhandled-exceptions-by-default.patch	1,82 ko	06 août 2021	Frédéric Péters
---	---------	--------------	-----------------