Publik - Development #67541

tests : faire concorder les cibles tox aux versions de debian (?)

20 juillet 2022 12:12 - Paul Marillonnet

Statut:	Nouveau	**Début:**	20 juillet 2022
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Club:**	Non
Planning:	Non		

Description

réflexions issues d'un ticket Authentic (#66488) où l'augmentation des restrictions sur des versions de dépendances dans tox donne une combinatoire dont on ne sait plus très bien quelles occurrences tester.
Genre faut-il tester la version A de django avec la version Y de django-tables 2, la version B de DRF avec la version Z de jwcrypto, etc.
À chaque fois, pour avoir la réponse il convient de se référer aux versions des paquets de buster, bullseye, bullseye-backports pour déterminer quels n-uplets de cette combinatoire sont pertinents.
Plutôt que d'avoir à faire ce travail pour déterminer quels sont les restrictions pertinentes à taper à chaque fois qu'on modifie tox, on pourrait partout nommer explicitement les cibles d'après les versions de debian.
Exemple, dans authentic encore, où :

```
$ tox -l
py3
py3-buster
py3-bullseye
py3-stable-backports
code-style
```

(py3 étant la cible sans aucune restriction, pour voir si les dernières versions tirées conviennent toujours)

Historique

#1 - 20 juillet 2022 12:13 - Paul Marillonnet

- Description mis à jour

#2 - 20 juillet 2022 13:58 - Frédéric Péters

Sur un temps donné mais qui devrait être court on peut se trouver à déployer sur deux versions, oldstable et stable (ici buster et bullseye).

Comme je comprends, ta troisième situation ("stable-backports") correspond dans le cas actuel à "bullseye mais en tirant django 3.2 des backports", c'est-à-dire "on anticipe qu'une fois qu'on en aura terminé avec buster, on ira vers là".

Perso je préférerais un travail à ne pas avoir à gérer trois combinaisons, c'est-à-dire être plus rapide sur les mises à jour.

Par ailleurs j'aime assez la distribution dans le temps de la découverte qu'une nouvelle version d'une dépendance va demander adaptation, plutôt que se trouver à ne rien voir jusqu'au moment où on souhaite monter de version debian et où on se tape en même temps les évolutions django et djangorestframework et django-tables2 etc.

Ce qui me ferait dire que, si on part sur multiplier les cibles, mes deux premières seraient :

- tox-current : correspond à notre unique environnement cible, aujourd'hui bullseye
- tox-next : correspondrait à l'environnement "+1", i.e. aujourd'hui bullseye + django 3.2 des backports

Une fois le taf réalisé et qu'on déploie sur bullseye + django 3.2, et en supposant qu'il n'y a pas déjà eu de nouvelle version debian, ça deviendrait :

- tox-current : bullseye + django 3.2
- tox-next : le moins de limite sur les versions

Arrive la publication de bookworm, (debian 12)

- tox-current : bullseye + django 3.2
- tox-next : bookworm

~~

Mais on parle de ça on ne va pas doubler tripler ou pire le temps de builds, donc j'imagine que ça va passer par de rares exécutions la nuit, et si on regarde la situation actuelle on voit que ces builds nocturnes ne sont pas corrigés. (depuis au moins 2 semaines l'historique affiché ne remonte pas plus loin pou authentic).

Donc je dirais qu'il y aurait à d'abord faire en sorte que sur le module où c'est plus ou moins appliqué ça fonctionne, avant de vouloir généraliser.

#3 - 20 juillet 2022 15:02 - Paul Marillonnet

Frédéric Péters a écrit :

> Sur un temps donné mais qui devrait être court on peut se trouver à déployer sur deux versions, oldstable et stable (ici buster et bullseye).
>
> Comme je comprends, ta troisième situation ("stable-backports") correspond dans le cas actuel à "bullseye mais en tirant django 3.2 des backports", c'est-à-dire "on anticipe qu'une fois qu'on en aura terminé avec buster, on ira vers là".
>
> Perso je préférerais un travail à ne pas avoir à gérer trois combinaisons, c'est-à-dire être plus rapide sur les mises à jour.

Ok, je comprends.

> Par ailleurs j'aime assez la distribution dans le temps de la découverte qu'une nouvelle version d'une dépendance va demander adaptation, plutôt que se trouver à ne rien voir jusqu'au moment où on souhaite monter de version debian et où on se tape en même temps les évolutions django et djangorestframework et django-tables2 etc.
>
> Ce qui me ferait dire que, si on part sur multiplier les cibles, mes deux premières seraient :
> […]

Perso tout me va à peu près tant qu'on a pas à maintenir des environnements du genre py3-django123-djtables456-drf890-jwcrypto123 etc.
Je n'avais pas pensé à ce schéma sur deux cibles qui notamment me va très bien.

> Mais on parle de ça on ne va pas doubler tripler ou pire le temps de builds, donc j'imagine que ça va passer par de rares exécutions la nuit, et si on regarde la situation actuelle on voit que ces builds nocturnes ne sont pas corrigés. (depuis au moins 2 semaines l'historique affiché ne remonte pas plus loin pou authentic).
>
> Donc je dirais qu'il y aurait à d'abord faire en sorte que sur le module où c'est plus ou moins appliqué ça fonctionne, avant de vouloir généraliser.

Fair enough :)
Je regarde.

#4 - 20 juillet 2022 15:11 - Paul Marillonnet

Paul Marillonnet a écrit :

> Fair enough :)
> Je regarde.

#67550.