Publik Installation Développeur - Development #50525

Permettre l'utilisation de uwsgi (en plus du serveur django)

25 janvier 2021 17:31 - Emmanuel Cazenave

Statut: Fermé Début: 25 janvier 2021

Priorité: Normal Echéance:

Assigné à: Emmanuel Cazenave % réalisé: 0%

Catégorie: Temps estimé: 0:00 heure

Version cible:

Patch proposed: Oui Planning: Non

Description

Uwsgi utile pour tester les jobs (#48407, #50017), le serveur django restant utile pour aller taper du pdb etc.

Et parait-il qu'il y a aurait moyen de faire un truc intelligent qui tente d'abord le serveur uwsgi et qui se rabat su le serveur django si jamais je sais pas quoi, et que tout ceci marcherait encore plus facilement en utilisant systemd plutôt que supervisord.

Je laisse l'auteur de cette idée disruptive en expliquer les grandes lignes.

Demandes liées:

Lié à Combo - Development #52221: uwsgi, spooler, connexion db Fermé 19 mars 2021

Révisions associées

Révision 934cfff6 - 13 avril 2021 15:14 - Emmanuel Cazenave

add uwsgi as available application server (#50525)

Historique

#1 - 26 janvier 2021 01:26 - Thomas Noël

Emmanuel Cazenave a écrit :

Et parait-il qu'il y a aurait moyen de faire un truc intelligent qui tente d'abord le serveur uwsgi et qui se rabat su le serveur django si jamais je sais pas quoi

Cf https://nginx.org/en/docs/http/ngx http upstream module.html

En gros sans doute quelque chose de ce genre :

```
upstream combo {
    server 127.0.0.1:8034 weight=5;  # par défaut, supervisor / runserver
    server unix:/socket/uwsgi backup;  # mais si y'a pas, uwsgi
}
server {
    location / {
        proxy_pass http://combo;
    }
}
```

#2 - 26 janvier 2021 10:34 - Emmanuel Cazenave

Et systemd c'était comme ça ? Il me semblait que tu avais une bonne raison, dont je ne me rappelle plus.

#3 - 26 janvier 2021 10:40 - Thomas Noël

Emmanuel Cazenave a écrit :

Et systemd c'était comme ça ? Il me semblait que tu avais une bonne raison, dont je ne me rappelle plus.

J'ai un peu regardé systemd-user et ça me semble pas si simple, on pourrait "juste" basculer en uwsgi dans un premier temps, en modifiant supervisorctl. Mais tout à coup je me demande si uwsgi sait faire du reload-automatique-quand-je-touche-un-fichier-py à la runserver, et j'en doute, or c'est quand même ça qui est super pratique avec runserver.

Peut-être que mon idée disruptive n'est pas si géniale.

29 avril 2024 1/4

#4 - 26 janvier 2021 10:48 - Emmanuel Cazenave

Thomas Noël a écrit :

Mais tout à coup je me demande si uwsgi sait faire du reload-automatique-quand-je-touche-un-fichier-py à la runserver, et j'en doute, or c'est quand même ça qui est super pratique avec runserver.

Ça me dérange pas, d'autant qu'on resterait sur du runserver par défaut, ce serait quand même bien d'avoir uwsgi sous la main prêt à l'emploi.

#5 - 26 janvier 2021 10:50 - Benjamin Dauvergne

Thomas Noël a écrit :

J'ai un peu regardé systemd-user et ça me semble pas si simple, on pourrait "juste" basculer en uwsgi dans un premier temps, en modifiant supervisorctl. Mais tout à coup je me demande si uwsgi sait faire du reload-automatique-quand-je-touche-un-fichier-py à la runserver, et j'en doute, or c'est quand même ça qui est super pratique avec runserver.

Ça existe, https://uwsgi-docs.readthedocs.io/en/latest/Options.html#py-auto-reload

#6 - 23 mars 2021 11:16 - Emmanuel Cazenave

- Lié à Development #52221: uwsgi, spooler, connexion db ajouté

#7 - 07 avril 2021 17:03 - Emmanuel Cazenave

- Fichier 0001-add-uwsgi-as-available-application-server-50525.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Assigné à mis à Emmanuel Cazenave
- Patch proposed changé de Non à Oui

Voilà qui permet d'utiliser uwsgi, je n'en fais pas le serveur par défaut pour autant, laissons mûrir.

Pour passer de runserver à uwsgi, c'est par exemple pour combo :

```
supervisorctl stop combo
supervisorctl start combo-uwsgi
```

Les deux service sont configurés pour écouter sur le même port, pas de cohabitation possible, mais du coup rien à changer cote nginx.

Dans les chose notables, je réutilise les uwsgi.ini inclus dans les sources, modulo quelques modifications (la socket, autoreload ...).

Aussi je tape un lien symbolique du uwsidecorators système dans le virtualenv, le uwsgidecorators de pypi est pas du tout à jour, et de toute façon ça sent les emmerdes de faire autrement.

#8 - 07 avril 2021 17:31 - Emmanuel Cazenave

Emmanuel Cazenave a écrit :

Dans les chose notables, je réutilise les uwsgi.ini inclus dans les sources, modulo quelques modifications (la socket, autoreload ...).

J'oubliais le point important des cron uwsgi qui sont shootés. Je suis une peur frileux, ça pourra être revu, en attendant il y a déjà possibilité de surcharger la conf uwsgi dans des ~/.config/publik/settings/foo/settings.d/uwsgi-local.ini pour les amateurs.

#9 - 07 avril 2021 17:39 - Thomas Noël

Emmanuel Cazenave a écrit :

Emmanuel Cazenave a écrit :

Dans les chose notables, je réutilise les uwsgi.ini inclus dans les sources, modulo quelques modifications (la socket, autoreload ...).

J'oubliais le point important des cron uwsgi qui sont shootés. Je suis une peur frileux, ça pourra être revu, en attendant il y a déjà possibilité de surcharger la conf uwsgi dans des ~/.config/publik/settings/foo/settings.d/uwsgi-local.ini pour les amateurs.

(rien à voir : si c'est toujours pour l'affaire de mails qui pourraient partir par erreur, je me disais l'autre jour qu'on pourrait intégrer https://docs.djangoproject.com/en/2.2/topics/email/#file-backend dans devinst ... mais d'abord voir comment gérer ça dans wcs ... encore un fil à tirer d'une pelote de laine)

29 avril 2024 2/4

#10 - 07 avril 2021 18:13 - Frédéric Péters

Bout de pelote qu'on regarde sans tirer : #36977

#11 - 08 avril 2021 00:50 - Thomas Noël

Frédéric Péters a écrit :

Bout de pelote qu'on regarde sans tirer : #36977

Ca me rappelait bien quelque chose, et je vais pas me relancer dans la bataille, mais sans doute que devinst a évolué depuis hein bon.

Bref, impeccable pour ce patch déjà Manu, et allons-y déjà sans les cron, on verra ensuite. Déjà les spooler c'est cool.

Je me dis qu'on pourrait quand même dès maintenant modifier les uwsgi.ini pour y être moins bourrins en lancement de workers :

- spooler-processes = 1 (au lieu de 3 par défaut, 1 suffirait en local mais avec 2 on peut voir les pépins de)
- processes = 5 (parce qu'on a mis 500 sur nos machines mais bon en local, bon)
- cheaper = 2 (nombre mini de workers)
- cheaper-initial = 2 (worker au démarrage)
- cheaper-step = 1 (on ajoute des workers un par un)
- cheaper-busyness-backlog-alert = 2 (workers en cas d'urgence)
- cheaper-busyness-backlog-step = 2 (workers en cas d'urgence)
- cheaper = 3

Et aussi, il doit manquer dans un coin la création des répertoires spooler, non ? (uwsgi ne le fait pas de lui même)

Dans la config de supervisor je n'ai pas compris le commentaire « ; Concurrency set to 1 because there is no lock around calls to hobo_notify », c'est peut-être juste une scorie de hobo-agent à supprimer ?

Aussi, plutôt mettre les cron en commentaire ? Genre avec

```
ansible.builtin.replace:
  path: "{{uwsgi_settings}}"
  regexp: '^(cron .*)$'
  replace: '#commented by devinst# \1'
```

#12 - 13 avril 2021 15:46 - Emmanuel Cazenave

- Fichier 0001-add-uwsgi-as-available-application-server-50525.patch ajouté

Thomas Noël a écrit :

Je me dis qu'on pourrait quand même dès maintenant modifier les uwsgi.ini pour y être moins bourrins en lancement de workers

On pourrait attendre de voir à l'usage ? C'est censé s'adapter à une faible charge, je suis partisan d'attendre une éventuelle constatation empirique que c'est chiant ce nombre de worker excessif au début avant de s'embêter à tout changer.

Et aussi, il doit manquer dans un coin la création des répertoires spooler, non ? (uwsgi ne le fait pas de lui même)

Fait, j'avais zappé.

Dans la config de supervisor je n'ai pas compris le commentaire « ; Concurrency set to 1 because there is no lock around calls to hobo_notify », c'est peut-être juste une scorie de hobo-agent à supprimer ?

Oui, supprimé.

Aussi, plutôt mettre les cron en commentaire ? Genre avec

Fait (avec une syntaxe légèrement différente parce la doc pointée concerne une version d'ansible trop récente par rapport à ce que l'on a dans debian). Et ça m'a donné l'idée de #53066.

#13 - 13 avril 2021 15:53 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Emmanuel Cazenave a écrit :

29 avril 2024 3/4

Thomas Noël a écrit :

Je me dis qu'on pourrait quand même dès maintenant modifier les uwsgi.ini pour y être moins bourrins en lancement de workers

On pourrait attendre de voir à l'usage ? C'est censé s'adapter à une faible charge, je suis partisan d'attendre une éventuelle constatation empirique que c'est chiant ce nombre de worker excessif au début avant de s'embêter à tout changer.

On sera quand même sur 5 workers par brique au moins (cheaper = 5), ça va bouffer de la ram gratuitement. Bon après oui, on s'en fout, on a plein de RAM sur nos machines persos, on verra à l'usage ça me va très bien !

#14 - 13 avril 2021 16:20 - Emmanuel Cazenave

- Statut changé de Solution validée à Solution déployée

commit 934cfff69dff3ea569cb547a017ac8b40dfdcc89
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Wed Mar 31 16:22:35 2021 +0200

add uwsgi as available application server (#50525)

Fichiers

0001-add-uwsgi-as-available-application-server-50525.patch	9,81 ko	07 avril 2021	Emmanuel Cazenave
0001-add-uwsgi-as-available-application-server-50525.patch	9,82 ko	13 avril 2021	Emmanuel Cazenave

29 avril 2024 4/4