

Hobo - Development #16599

cook : vérification des noms de domaine avant de lancer le déploiement

29 mai 2017 16:00 - Frédéric Péters

Statut:	Fermé	Début:	29 mai 2017
Priorité:	Normal	Echéance:	
Assigné à:	Christophe Siraut	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Si le nom de domaine ne résoud pas, le déploiement partira dans le vide, autant détecter ça dès que possible.			
Demandes liées:			
Lié à Hobo - Development #23823: Fournir une API de vérification de bon fonct...		Fermé	14 mai 2018
Lié à Hobo - Bug #32687: regression sur cook avec plusieurs variables		Fermé	29 avril 2019

Révisions associées

Révision 73848f82 - 16 avril 2019 18:05 - Christophe Siraut

commands/cook: add prechecks on recipe (#16599)

Historique

#1 - 23 juillet 2017 11:13 - Thomas Noël

En fait on pourrait aller jusqu'à tester si les URL à déployer ne répondent pas 404 : dans ce cas, message d'erreur et arrêt. (si c'est pas 404 c'est que le logiciel qui doit faire le déploiement n'est pas là)

#2 - 05 juin 2018 15:45 - Serghei Mihai

Yep, et dans le message d'échec de déploiement rajouter la raison: nom de domaine inexistant, reponse HTTP foireuse, erreur de certificat.

#3 - 05 juin 2018 15:53 - Christophe Siraut

- Lié à Development #23823: Fournir une API de vérification de bon fonctionnement ajouté

#5 - 13 septembre 2018 11:29 - Christophe Siraut

- Fichier 0002-cook-add-prechecks-on-recipe-16599.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

proposition en attente de [#23823](#)

on pourrait aller jusqu'à tester si les URL à déployer ne répondent pas 404

je n'ai pas été jusque là (je me dis que c'est probablement suffisant en l'état)

#6 - 13 septembre 2018 12:09 - Christophe Siraut

- Fichier 0002-cook-add-prechecks-on-recipe-16599.patch ajouté

- ajouté une condition pour les actions sans url.

#7 - 13 septembre 2018 15:01 - Thomas Noël

Christophe Siraut a écrit :

- ajouté une condition pour les actions sans url.

(c'est le même patch)

#8 - 13 septembre 2018 15:20 - Christophe Siraut

- Fichier 0002-cook-add-prechecks-on-recipe-16599.patch ajouté

- Statut changé de *Solution proposée* à *En cours*

(c'est le même patch)

bizarre (une fenêtre ouverte dans nautilus au contenu périmé, le contenant est un lien symbolique) voici.

#9 - 14 septembre 2018 14:40 - Christophe Siraut

- Fichier 0002-cook-add-prechecks-on-recipe-16599.patch ajouté

- Statut changé de *En cours* à *Solution proposée*

version simplifiée (moins de code)

#10 - 14 septembre 2018 15:08 - Thomas Noël

Typo : s/resovable/resolvable/

Et ça serait mieux de raiser un `CommandError` plutôt qu'un `sys.exit`.

Et c'est juste moi, mais je préfère cette forme pour gérer l'absence d'url dans l'action:

```
if 'url' not in action_args:
    continue
...
```

Mais pour moi, ça marche pas avec des déclarations comme "url": "https://\${combo}" qu'on pose dans nos `recipe.json`. Il faut gérer les variables d'abord.

#11 - 17 septembre 2018 13:21 - Christophe Siraut

- Fichier 0002-cook-add-prechecks-on-recipe-16599.patch ajouté

Il faut gérer les variables d'abord.

Oui c'est juste, ajouté une méthode `get_steps()` pour compiler les arguments.

#12 - 17 septembre 2018 16:00 - Thomas Noël

Christophe Siraut a écrit :

Il faut gérer les variables d'abord.

Oui c'est juste, ajouté une méthode `get_steps()` pour compiler les arguments.

Je trouve pas ça super joli (une méthode appelée une seule fois, je doute toujours ; et le patch serait plus lisible sans). Je sais, je suis pénible, mais j'ai pas envie que cette partie du code devienne trop touffue...

#13 - 03 octobre 2018 18:20 - Christophe Siraut

- Fichier 0001-commands-cook-add-prechecks-on-recipe-16599.patch ajouté

simplification.

#14 - 02 mars 2019 11:19 - Benjamin Dauvergne

- Statut changé de *Solution proposée* à *En cours*

- Assigné à mis à Christophe Siraut

Pas bon le découpage de la boucle for step in a perdu des bouts au passage, je ferai une première passe d'assemblage des action/arction_args, puis éventuels checks(), puis application, comme cela c'est bien découpé (j'aurai préféré des méthodes check_<action>()) mais comme on a autant de méthodes que de type de briques, ça attendra une évolution quand on aura une action avec un paramètre url qui n'est pas le déploiement d'une brique).

#15 - 15 mars 2019 14:11 - Christophe Siraut

J'ai adapté le patch en suivant la remarque de Benjamin.

#16 - 15 mars 2019 15:19 - Benjamin Dauvergne

Il me semble qu'on a aucun test sur cook...

#19 - 19 mars 2019 19:00 - Christophe Siraut

- Statut changé de En cours à Solution proposée

J'ai ajouter deux tests sommaires, ça me semble suffisant (sinon il faut mocker les différentes fonctions réseau pour éviter par exemple de faire des vérification de certificat en vrai à chaque exécution des tests)

#22 - 21 mars 2019 09:15 - Christophe Siraut

Déplacé les tests vers le bon dossier. (Et je ne suis pas parvenu à concilier pytest-django et django-tenant-schema)

#23 - 21 mars 2019 09:42 - Benjamin Dauvergne

Christophe Siraut a écrit :

Déplacé les tests vers le bon dossier. (Et je ne suis pas parvenu à concilier pytest-django et django-tenant-schema)

Sans charger debian_config_common c'est difficile effectivement, il suffirait de l'ajouter à tests/settings.py comme on le fait pour multitenant (le fait est que hobo le charge aussi dans debian/server/debian_config.py).

#26 - 27 mars 2019 16:30 - Frédéric Péters

C'est un truc que je disais à Nicolas par rapport à l'intégration Matomo je préfère garder le maximum hobo indépendant du côté multitenants (comme les autres applications).

Tester un bout de code vérifiant une série de noms de domaine, ça me semble pouvoir tenir sans faire appel à du multitenant.

(qu'il y ait, par ailleurs, de nouveaux tests appelant toute l'infrastructure multitenant pour un gros test de cook, très bien).

Dans la pratique, ça m'ennuie de voir un changement comme :

```
-def test_response(app, admin_user, services, monkeypatch):  
+def test_response(tenant_schema, fake_notify, app, admin_user, services, monkeypatch):
```

qui me fait dire que tous les tests devraient désormais dépendre de deux fixtures supplémentaires.

#27 - 27 mars 2019 17:00 - Christophe Siraut

- Statut changé de Solution proposée à Information nécessaire

qui me fait dire que tous les tests devraient désormais dépendre de deux fixtures supplémentaires.

Une seule fixture supplémentaire : tenant_schema. (L'autre c'était une erreur de ma part)

je préfère garder le maximum hobo indépendant du côté multitenants

C'est bête j'ai travaillé à convertir les tests existants pour fonctionner avec tenant-schemas. En pratique les installations que je connais utilisent majoritairement cette configuration (et je voulais éviter d'ajouter un namespace "test_schemas" supplémentaire). Préfères-tu abandonner bb73ce7669559236887d19a7350f4797d4b54526 ou doubler ces tests : avec et sans tenant-schemas?

#28 - 27 mars 2019 17:11 - Frédéric Péters

Je confirme mon commentaire précédent, je préfère garder le maximum indépendant du multitenant. (et ça inclut les tests).

#29 - 28 mars 2019 10:14 - Benjamin Dauvergne

Je n'ai pas d'avis sur les remarques de Fred, de mon côté ça m'intéressait juste de ne pas mélanger les tests de hobo avec les tests de l'infra multitenant; si seul la commande hook demande le mélange des deux, je ne pense pas que ça pose problème que seul ses tests dépendent d'une fixture avec multitenant.

#30 - 28 mars 2019 10:41 - Christophe Siraut

- Statut changé de Information nécessaire à Solution proposée

J'ai adapté le patch en séparant les tests qui utilisent tenant-schema.

#31 - 16 avril 2019 12:17 - Nicolas Roche

J'ai relu.

Peut-être ajouter les 2 tests ci-dessous pour couvrir tout le code de la fonction ajoutée.

```
def test_cook_invalid_certificat(db, fake_notify, monkeypatch):
    monkeypatch.setattr(ServiceBase, 'is_resolvable', lambda x: True)
    monkeypatch.setattr(ServiceBase, 'has_valid_certificate', lambda x: False)
    with pytest.raises(CommandError) as e_info:
        call_command('cook', 'tests_schemas/recipe.json')
        assert 'has no valid certificate' in str(e_info.value)

def test_cook_unknown_action(db, fake_notify, monkeypatch):
    RECEIPE="""
{
  "steps": [
    {"create-Bobo": {
      "url": "https://dev.entrouvert.org/"
    }}
  ]
}
"""
    recipe_file = '/tmp/recipe_having_wrong_action.json'
    with open(recipe_file, 'w') as handler:
        handler.write(RECEIPE)
    with pytest.raises(CommandError) as e_info:
        call_command('cook', recipe_file)
        assert 'Unknown action Bobo' in str(e_info.value)
```

Voir si on peut pas plutôt tester `Command.check_action()` dans 'tests', mais garder quand même 'tests_schemas' pour les futurs tests sur les déploiements via cook. Je regarde si je peux proposer quelque-chose dans ce sens...

#32 - 16 avril 2019 14:58 - Nicolas Roche

- Fichier test_cook.py ajouté

- Statut changé de Solution proposée à En cours

Je vous propose d'ajouter ce fichier dans le répertoire '/tests'.

Mais je suis également preneur pour un nouveau répertoire '/tests_schemas' pour les test à venir sur les déploiements pré-configurés qui nécessiteraient d'avoir un vrai objet `Command` sous la main (comme par exemple [#14630](#)).

#33 - 16 avril 2019 15:40 - Christophe Siraut

- Statut changé de En cours à Solution proposée

J'ai inclus ton fichier de tests (plus clair, merci) en sortant les assert des blocs `pytest.raises()`.

#34 - 16 avril 2019 17:14 - Nicolas Roche

- Statut changé de Solution proposée à Solution validée

Je valide.

#35 - 16 avril 2019 18:06 - Christophe Siraut

- Statut changé de Solution validée à Résolu (à déployer)

```
commit 73848f82f3ac0eeca52b2e1891b4e15d4019c970 (HEAD -> master, origin/master, origin/HEAD)
```

Author: Christophe Siraut <csiraut@entrouvert.com>

Date: Wed Mar 27 17:34:01 2019 +0100

commands/cook: add prechecks on recipe (#16599)

#36 - 19 avril 2019 11:16 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

#37 - 29 avril 2019 19:31 - Nicolas Roche

- Lié à Bug #32687: regression sur cook avec plusieurs variables ajouté

Fichiers

0002-cook-add-prechecks-on-recipe-16599.patch	3,16 ko	13 septembre 2018	Christophe Siraut
0002-cook-add-prechecks-on-recipe-16599.patch	3,16 ko	13 septembre 2018	Christophe Siraut
0002-cook-add-prechecks-on-recipe-16599.patch	3,25 ko	13 septembre 2018	Christophe Siraut
0002-cook-add-prechecks-on-recipe-16599.patch	3,08 ko	14 septembre 2018	Christophe Siraut
0002-cook-add-prechecks-on-recipe-16599.patch	4,88 ko	17 septembre 2018	Christophe Siraut
0001-commands-cook-add-prechecks-on-recipe-16599.patch	2,91 ko	03 octobre 2018	Christophe Siraut
test_cook.py	2,29 ko	16 avril 2019	Nicolas Roche