

Supervision Drupal

Frédéric VACHON

<frederic.vachon@savoirfairelinux.com>

13 juillet 2015

1 Besoin de supervision Drupal

1.1 Définition du besoin

L'équipe qui travaille avec Drupal avait besoin d'une solution de supervision multi-site. L'idée est d'avoir un espace centralisé où il serait possible d'avoir un aperçu de l'état de santé de plusieurs sites d'un coup d'oeil. Il faut aussi avoir accès à une vue détaillée de chacun des sites.

1.2 Analyse du besoin

Un des outils déjà utilisé par les administrateurs Drupal offre les fonctionnalités nécessaires à la cueillette de données sur les sites Drupal. Cet outil s'appelle *site audit* et s'intègre à un autre outil ligne de commande qui s'appelle *drush*. *Site audit* génère des rapports à propos de plusieurs catégories d'informations sur un site Drupal. Quant à *drush*, il contient quelques fonctionnalités qui pourront être utilisées pour aller chercher des informations que *site audit* ne fournit pas, comme s'il y a des mises à jour disponibles pour le *Core* et pour les autres modules.

Quant au système de supervision à utiliser, l'équipe de supervision chez SFL utilise *Alignak*. L'équipe de supervision développe actuellement un API REST qui permettra d'offrir un service de supervision. Elle travaille aussi au développement d'une interface web pour présenter les données collectées par le système de supervision. Ces deux technologies, *Surveil* et *Bansho* respectivement, seront utilisées pour notre solution de supervision Drupal.

2 Livrables

2.1 Plugins

La composante principale de la solution de supervision Drupal est la suite de plugins. Les plugins sont utilisés par *Alignak* pour amasser des informations de supervision sur un hôte donné. La solution est basée sur 12 plugins lesquels se trouvent à cet URL :

`https://github.com/savoirfairelinux/monitoring-tools/`

Voici la liste des plugins :

- `check_drupal_cache` : Récupère de l'information sur l'état de la cache
- `check_drupal_codebase` : Récupère de l'information sur la qualité du code
- `check_drupal_cron` : Récupère de l'information sur les cron jobs
- `check_drupal_database` : Récupère de l'information sur l'état de la base de données
- `check_drupal_extensions` : Récupère de l'information sur les extensions
- `check_drupal_jenkins` : Récupère le statut du dernier build du projet
- `check_drupal_logging` : Récupère le status de la journalisation
- `check_drupal_security` : Récupère des informations concernant la sécurité du site
- `check_drupal_status` : Récupère de l'information concernant les mises à jour disponibles, le nombre de modules activés, la version du core, de php, de la base de données et le SGBD utilisé.
- `check_drupal_views` : Récupère des informations sur le module Views
- `check_http_load` : Effectue un test de charge sur le site

- `check_selenium` : Exécute un scénario sur la page Drupal et retourne le temps total

2.2 Pack

Un pack est un regroupement de configuration pour *Alignak*, qui permet de largement simplifier la configuration de son système de supervision. Pour la solution de supervision de Drupal, un pack a été créé. Celui-ci permet de configurer le système de supervision en lançant les deux commandes suivantes :

```
$ surveil-pack-upload -mongo-uri MONGODB_URI PATH_TO_PACK
```

```
$ surveil config-host-create -host-name HOST_NAME -address HOST_ADDRESS -use linux-drupal
```

La première commande charge le pack dans la base de données, puis la commande `-use` spécifie le pack à utiliser pour configurer l'hôte. Il est possible de configurer les plugins en utilisant l'option `-custom-fields` qui permet de choisir les arguments à passer aux plugins.

La documentation du pack est disponible à cet URL :

```
http://github.com/savoirfairelinux/monitoring-tools/blob/master/  
packs/linux-drupal/doc/linux-drupal.rst
```

Plus d'informations sur les packs sont disponibles à cet URL :

```
http://shinken.readthedocs.org/en/latest/14_contributing/packs.html
```

2.3 Packages

Tous les plugins ont été *packagés* afin d'en faciliter l'installation. Les plateformes cibles sont :

- CentOS 7
- Debian 8
- Ubuntu 12.04
- Ubuntu 14.04

Les *packages* sont construits en utilisant le service Joulupukki qui est hébergé dans l'infrastructure locale à l'URL suivant :

`http://packager.savoirfairelinux.net`

2.4 Module pour l'interface web

La couche présentation de la solution est intégrée dans une interface web maison actuellement en développement : *Bansho*. Il s'agit d'un module qui sera accessible via l'URL suivant :

`http://mon.hote/#/view?view=drupalDashboard`

2.5 Dépôt Drupal monitoring

Le développement de la solution s'est fait à l'aide du dépôt Drupal monitoring. Il contient l'ensemble des technologies nécessaires pour faire fonctionner la solution. Ce dépôt constitue l'exemple auquel se référer pour comprendre toutes les technologies qui sont utilisées et comment celles-ci interagissent entre elles. Il est aussi possible d'utiliser ce projet pour déployer une solution toute-en-une. Les étapes sont documentées sur sa page Github :

<https://github.com/savoirfairelinux/drupal-monitoring>

Cette solution n'est appropriée que pour les utilisateurs qui n'ont pas de système de supervision avec *Surveil* et *Bansho* à leur disposition. Il s'agit néanmoins de la manière la plus simple d'utiliser la solution. Il est à noter que ce dépôt constitue la documentation principale de la solution.

3 Déploiement

Puisque Savoir-faire Linux est actuellement en train d'implanter une nouvelle architecture locale, le déploiement de la solution de supervision Drupal ne peut être effectué dès maintenant. Lorsque l'implantation sera terminée, *Surveil* et *Bansho* y seront déployés, ce qui permettra le déploiement de la solution.

Il suffira d'utiliser *Surveil* pour ajouter le pack mentionné plus haut et configurer le système de supervision via ce pack. Les plugins devront être installés sur la machine où *Alignak* tournera. Il sera alors possible de consulter l'interface web en utilisant l'URL mentionné ci-dessus.

Puisque les plugins utilisent *drush* à distance avec les alias, la machine qui va lancer les *checks* devra avoir *drush* d'installé. Elle devra également partager sa clé publique avec chacune des machines hébergeant les sites Drupal supervisés. Les alias devront également être configurés sur la machine lançant les *checks*. Dans le cas où la configuration des alias est constamment changeante, un script devra s'occuper de récupérer ces configurations afin que les *checks* utilisent toujours une configuration à jour des alias.