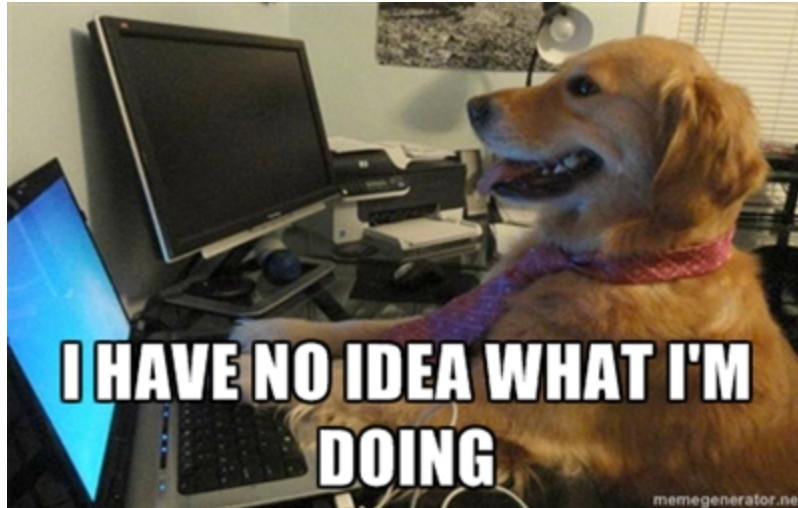


Stage DevOps - Test technique



Abstract

Le but de ce test est de juger vos capacités d'adaptation et vos compétences techniques. Ce sujet, bien que relativement simple, cache quelques pièges qu'il vous faudra savoir déjouer.

Vous allez devoir réaliser une application web automatisant le lancement d'un script que vous allez écrire. Vous devrez également mettre en place vous même votre infrastructure.

Vous avez une (petite) semaine pour réaliser le projet (ce qui est largement suffisant puisque je l'ai fait en moins de 2h).

Index

[Abstract](#)

[Index](#)

[1. Pré-requis](#)

[2. Modalités de rendu](#)

[3. Administration système](#)

[3.1. Stack web](#)

[3.1.1. Nginx](#)

[3.1.2. Gunicorn](#)

[3.1.3. Supervisord](#)

[3.2. MySQL](#)

[4. Scripting](#)

[5. Application web](#)

[5.1. Formulaire de lancement](#)

[5.2. Listing](#)

[5.3. Rendu](#)

1. Pré-requis

Le sujet est à réaliser chez vous (ou ailleurs en fait) sur une image de VM que nous fournissons (et que nous utiliserons aussi pour corriger votre rendu). Libre à vous de faire autrement mais nous vous conseillons de mettre toutes les chances de votre côté.

Téléchargez l'image de la VM disponible sur <http://misc.quanta.gr/recrutement.tgz> et faites la fonctionner chez vous pour vos tests.

Il vous faudra un outil de virtualisation susceptible de faire fonctionner la VM (VMware Player/Fusion ou Virtualbox devraient faire l'affaire).

Vous pouvez vous y connecter en SSH ou sur la console avec l'identifiant *root* et le mot de passe *quanta*.

Plus d'infos sur la VM si vous ne voulez pas l'utiliser mais seulement reproduire son comportement:

- OS: Debian 7.6
- `uname -a`: Linux recrutement 3.2.0-4-amd64 #1 SMP Debian 3.2.60-1+deb7u1 x86_64 GNU/Linux

2. Modalités de rendu

Vous devez rendre le projet avant le vendredi 25 juillet à 23h42.

Le rendu se fera par mail à jobs@quanta-computing.com, le sujet devra être "[STAGE][DEVOPS][TEST] Rendu <prenom>-<nom>" (nous ignorerons les mails ne respectant pas ce format).

Le mail devra avoir une pièce-jointe nommée <prenom>-<nom>.tar.gz.

Le dossier racine extrait lors du lancement de la commande `tar -xf tarball.tar.gz` doit s'appeler "rendu", tout vos fichiers et dossiers doivent se trouver dans ce dossier.

Nous vous recommandons d'ajouter dans le contenu du mail les différents problèmes que vous avez rencontrés et comment vous les avez résolus (ou ce que vous avez tenté s'ils ne sont pas résolus)

Soyez bien rigoureux dans le respect des formats et des noms de fichier, vous serez corrigés par un être humain¹ mais ce sera un script qui s'occupera de mettre en place votre rendu sur notre VM de test.

3. Administration système

Tout au long de ce sujet vous allez être amené à effectuer différentes opérations sur le système. Si vous installez des paquets, vous devez en renseigner la liste dans le fichier *packages.txt* à la racine de votre rendu (un nom de paquet par ligne). Vous devez rendre le fichier même s'il est vide. (notre script fera un *apt-get install `cat packages.txt` htop*)²

3.1. Stack web

Vous devez installer une stack web et la configurer pour faire fonctionner votre futur application web. Cette stack sera composée de Nginx, Gunicorn et Supervisor (et de Django évidemment)

3.1.1. Nginx

Nginx sera chargé de délivrer le contenu statique de votre site et de forwarder les autres requêtes à Gunicorn.

Vous devez configurer un VirtualHost pour votre application, les informations suivantes peuvent vous être utiles:

- Votre application doit être accessible via l'URL <http://sshstats.recrutement.quanta.gr> (bien entendu, vous ne pourrez pas modifier nos DNS donc pour que cela fonctionne chez vous il faudra que vous ajoutiez une entrée dans votre fichier hosts pour tester cela)
- Votre application sera installée dans */var/www/sshstats/*
- Les logs d'accès et d'erreur seront à envoyer respectivement dans les fichiers */var/log/nginx/sshstats.recrutement.quanta.gr.access.log* et */var/log/nginx/sshstats.recrutement.quanta.gr.error.log*

Vous devrez rendre 2 fichiers dans un dossier *nginx* placé à la racine de votre rendu (veillez donc bien à ne modifier que ces fichiers):

- *nginx.conf*: Il remplacera le fichier */etc/nginx/nginx.conf* sur notre plate-forme de test
- *sshstats.recrutement.quanta.gr.conf* qui contient la configuration de votre virtualhost et sera placé dans le dossier */etc/nginx/sites-enabled/*

¹ Pour autant qu'on puisse en juger du moins...

² Bien que ce soit effectivement vachement amusant, "&& reboot ||" n'est pas un nom de paquet valide

3.1.2. Gunicorn

Installez Gunicorn. Selon la technique que vous choisissez, pensez à mettre à jour le fichier *packages.txt* ou le fichier *requirements.txt* (voir plus bas).

Il n'y a rien à rendre de spécifique pour cette partie³, veuillez juste à configurer correctement votre application pour qu'elle puisse utiliser Gunicorn.

3.1.3. Supervisord

Installez Supervisord.

Vous devez rendre le fichier *sshstats.conf* dans un dossier *supervisor* de votre répertoire de rendu.

Ce fichier sera placé dans */etc/supervisor/conf.d/* par notre script, il doit permettre de contrôler de votre application via la commande *supervisorctl* (par exemple *supervisorctl restart sshstats* doit relancer votre application).

3.2. MySQL

Installez un serveur MySQL et créez les ressources suivantes:

- Un utilisateur MySQL *sshstats* (avec le mot de passe *sshstats*)
- Une base de donnée *sshstats* (accessible par l'utilisateur MySQL *sshstats* avec toutes les permissions)

Vous avez le droit de modifier la configuration de votre serveur MySQL mais uniquement dans le fichier */etc/mysql/my.cnf*.

Vous devez rendre les fichiers suivants dans un dossier *SQL* à la racine de votre rendu:

- *sshstats.sql*: Ce fichier est un script SQL qui doit créer votre utilisateur et votre bases de données, il sera exécuté par notre script d'installation via la commande *mysql < sshstats.sql*
- *my.cnf* qui remplacera le fichier */etc/mysql/my.cnf* sur notre plate-forme de test

4. Scripting

Vous devez écrire un script *sshstats* qui va lire le fichier */var/log/auth.log* et qui affichera sur la sortie standard des statistiques de connexions SSH.

³ Vous trouvez pas ça pénible de scroller tout le temps pour lire des bêtises ?

Vous devrez afficher une ligne par utilisateur (valide) ayant tenté de se connecter en indiquant à chaque fois:

- L'username utilisé
- Le nombre de connections réussies
- Le nombre de connections échouées

Ces trois champs seront séparés par des virgules (,) et chaque ligne sera terminée par un retour a la ligne (`\n`).

Vous pouvez, au choix, réaliser ce script en Python ou en Perl.

Le script doit être exécutable et doit pouvoir être lance en faisant `./sshstats`.

Le script est a placer dans un dossier *script* a la racine de votre répertoire de rendu, il sera installe dans un dossier `/etc/scripts/` par notre script.

5. Application web

Vous devez désormais créer une application web très simple composée d'une seule page.

Cette page devra contenir les 2 parties suivantes ainsi qu'une image quelconque ou vous voulez sur la page.

Vous n'attacherez pas d'importance a la présentation, le but est juste que les informations soient présentes et que l'application fonctionne, le but n'est pas d'en faire un site vitrine.

5.1. Formulaire de lancement

Vous devez afficher un formulaire prenant en paramètre un champ texte (maximum 42 caractères) représentant un commentaire.

Ce formulaire doit insérer en base de données un timestamp et le contenu de ce champ texte ainsi que le retour de votre script (`/etc/scripts/sshstats`).

5.2. Listing

En dessous du formulaire, affichez un listing de tout les lancements du scripts effectues par votre application (en affichant a chaque fois le timestamp et le commentaire).

Vous afficherez également la sortie du dernier lancement sur la page.

5.3. Rendu

Cette application **doit** être écrite en Python en utilisant le framework Django.

L'application web doit être rendue dans un dossier *www* a la racine de votre répertoire de rendu. Notre script copiera récursivement le contenu de ce dossier dans */var/www/sshstats/*.

Si vous utilisez des packages python supplémentaires, vous devez les renseigner dans un fichier *requirements.txt* a la racine de de votre application web (notre script fera un *pip install -r requirements.txt*)

Avant de tester votre projet, notre script executera les commandes suivantes (dans cet ordre):

- *python manage.py syncdb*
- *python manage.py migratedb*
- *supervisorctl reload && supervisorctl restart sshstats*