

May 2023 -- Deploy on Jelastic

Nouvel environnement

Pour mon premier test, j'ai choisi un environnement contenant :

- un 'application server' `apache-ruby` (`nginx-ruby` plus cher !) avec une IP publique (pas d'accès SLB),
- un 'noeud sql' `postgres 15.2` avec accès SLB.

Son nom est `name` et l'adresse est :

```
name.jcloud-ver-jpe.ik-server.com
```

Déploiement de l'application

1. Dans Application Server, cliquer sur 'Add-ons' et lancer 'Git-push-deploy'. Renseigner :
 - l'url github : `https://github.com/MaTsou/rentapp.git` ,
 - le nom de la branche à déployer : `prod` ,
 - le user (login github) : `mazu@sfr.fr` ,
 - le personnel token (à générer pour l'occasion dans github !).
2. Sur le noeud 'déploiement' nouvellement créé, changer le mode de `développement` (défaut) à `production` .
3. Dans Application Server, cliquer sur 'web-ssh' et :

```
cd webroot/ROOT (cela conduit à la racine de mon app)
bundle update
bundle install
```

4. Dans 'Gestionnaire de déploiement', renseigner un nouveau projet (celui qu'on vient de déployer); j'ai l'impression que 'Git-push-deploy' ne fait pas son travail complètement... Sans ça, on peut avoir des erreurs 'le projet n'existe pas' quand on met à jour depuis git (même si ça marche).

Dès lors, les mises à jour sur github peuvent être déployées, soit manuellement (clic sur la flèche verte...) soit automatiquement (configuration du déploiement).

Dans Application Server, cliquer sur la roue dentée puis 'variables'. À cet endroit, on a accès aux variables d'environnement et on peut en créer.

Créer une première variable d'environnement `RAILS_ENV` avec la valeur `production` et relancer ce server.

Lien avec la base de données

À la création de l'environnement, on reçoit un mail de confirmation avec les identifiants du gestionnaire de base de données -- qui est accessible en cliquant sur 'open in browser' (pour le noeud bdd) :

```
host: nodexxxxname.jcloud-ver-jpe.ik-server.com
id: webadmin
pwd: PQKmln28921
```

Mettre à jour `config/database.yml` :

```
production:
  <<: *default
  database: rentApp_production
  username: webadmin
  password: <%= ENV["RENTAPP_DATABASE_PASSWORD"] %>
```

La configuration et la sécurisation de la base de données se fait grâce à des variables d'environnement (à définir comme RAILS_ENV) :

nom	valeur
DATABASE_URL	postgresql://nodexxxxxx (le host du mail)
RENTAPP_DATABASE_PASSWORD	PQKmln28921

Cryptage des infos sensibles

Dans le web-ssh, `/webroot/ROOT` , créer `config/master.key` avec un contenu identique à celui que j'ai en local (sur ma machine). Attention, ce fichier ne doit pas se retrouver sur github car il contient la clé en clair. Cette clé ne doit en aucun cas être perdue ! Elle permet de décrypter le fichier `config/credentials.yml.enc` qui, lui, passe par github (et est généré à la création d'une app. ou avec `rails credentials:edit` .

Alors,

```
rails db:create
rails db:migrate
```

works and accessing database through browser gives access to all the tables and content.
Great for webmaster purposes..

Last thing : configure the public IP to access the app