DOCUMENTATION AWS-DJANGO

Pierre-Victor Chaumier

Décembre 2014

1. launch instance

   - go on EC2 page
   - on left panel, click on instance
   - click on launch instance button
   - select distro
   - select instance type
   - choose a security key or create a new one (DO NOT LOOSE IT, it is used to connect to the instance. Key lost means having to kill the instance and start again..) I put it in the /.ssh/ folder

   Now your instance should be up running

2. Create a DB

   - on the main AWS panel, go on RDS
   - click on get started now
   - Select database
   - production (be careful dafault one is not free)
   - choose details and settings (for instance my DB instance identifier is "dbinstance", my master username is "dbadmin" and my password is "password" no kidding i
   - complete advanced settings. BE CAREFUL with publicly accessible. I put no so that only my instances on AWS can access it. Choose a dbname (mine is dbname_of_project)
   - click on launching db instance. Creating a DB can take a little time so be patient.
   - create a security group (they are available from the ec2 dashboard on the left) name it as you wish (i choose dbsecuritygroup), inbound should be HTTP with custom IP for the source. The custom IP will be the identifier of the security group of your instance sg...

That is it, your DB should be safe and running now.

3. Setting up the shortcut to connect with ssh using config file in .ssh/ folder
    - go in your /.ssh/ folder
    - create a file named config
    - fill it as follow : Host name_of_shortcut HostName addresse of your instance ec2... User name_of_user (ubuntu for ubuntu instances) IdentityFile /path/to/key/file/ (generated earlier)
    - change rights chmod 700 for /.ssh/ folder chmod 600 for the key
    - run commande : ssh name_of_shortcut to connect to the instance

4. installation (postgre etc..) simply run the following commands (for debian/ubuntu instance):
    - sudo apt-get update
    - sudo apt-get upgrade
    - sudo apt-get install -y python git python-pip postgresql postgresql-server-dev-all python-dev libpq-dev supervisor nginx vim curl ntp libncurses5-dev make build-essential libssl-dev zlib1g-dev libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm tcl8.5

5. installation pyenv et pyenv virtualenv Normally, git should be installed by now. It is time to install pyenv and pyenv wrapper. Here is a good start. However, do not install following what is in the link but what is above.
    - To install pyenv, follow the very well made guide here : `https://github.com/yyuu/pyenv#installation` (source the .bash_profile file after modification or it might not be taken into account)
    - now install virtualenv following instructions here : `https://github.com/yyuu/pyenv-virtualenv`
    - create a virtualenv with this command

6. installer pip et migrations

7. nginx + gunicorn + supervisor

8. security groups pour donner accès a tout le monde au truc et à l'instance pour la DB