

Informe de la práctica de Laboratorio N°1: Instalación y configuración del entorno de trabajo para Django y Ruby on Rails.

Autor: Adán Rafael López Lecuona.

-Para empezar tuvimos que crear una nueva máquina virtual por defecto.

-Configuramos la red determinando los puertos como: -Rails ser ver|TCP|3000|3000 .

-SSH|TCP|2222|22.

Para comprobar que podemos comunicarnos entre ambos equipos mediante los puertos ya especificados anteriormente hacemos: -nc -l 3000 en la máquina virtual y nc -localhost 3000 en la máquina anfitriona.

Comprobamos el ssh : sudo apt-get install openssh_server. En el cliente realizamos ssh-p 2222 localhost para conectarnos con la máquina virtual.

Realizamos una clave situada en la carpeta ssh, siendo la id_rsa la clave privada y la rsa_pub la clave pública. Después de eso, tecleamos:

-ssh -keygen -t rsa .Genera la clave en el /ssh. Entonces copiamos la clave pública.

Cambiamos los permisos del ssh en la máquina virtual a 755. Todo esto para que el cliente pueda entrar sin clave .

Luego montamos la imagen VboxGuestAdditions.iso situada /sistemas-archivos/usr/share/virtualbox.

También montamos el dispositivo VboxLinuxAdditions en /mnt y comprobamos lo montado con el comando df -h . Instalándose en /mnt el archivo que añade módulos de kernel.

Ahora configuramos una carpeta compartida en la máquina anfitriona escogemos una y la hacemos permanente. En la máquina virtual realizamos:

-mount -t vboxsf dsi /home.

-mount -t vboxsf DSI -o uid=1000,gid=1000, /home/alu3954/DSI.

Tuve problemas al montar la carpeta compartida al no coincidir algunas opciones del commando.

Una vez configurado realizamos la segunda parte que es la instalación de Ruby on Rail en nuestro sistema operativo Ubuntu siguiendo una serie de pasos especificados en la página:

<http://gorails.com/setup/ubuntu/13.10>

Instalando primero sus dependencias más importantes. Luego configuramos el control de versiones que utilizaremos llamado Git, la configuración es única por cada usuario(nombre,email,clave). Finalmente estaremos autenticados en Github.

A la hora de configurar la base de datos configuré para que esté de base de datos por defecto Sqlite3 y de segunda base de datos utilizaré MySQL.

También tuvimos que cambiar la configuración del Gnome terminal marcando en título y comandos chequeando la casilla de: ejecutar comando como login shell.

Luego ejecutamos el script : `sudo chmod +x rails-installer.sh`.

<https://github.com/rkijha/RailsOnUbuntu/blob/master/rails-installer.sh>, ahí lo meteremos en nuestra carpeta compartida. Luego ejecutamos el cript en la máquina virtual: `./rail-installer.sh`

Con este comando creamos nuestra primera aplicación:

```
$ rails new first_app
create
create README.rdoc
create Rakefile
create config.ru
create .gitignore
create Gemfile
create app
create app/assets/javascripts/application.js
create app/assets/stylesheets/application.css
create app/controllers/application_controller.rb
.
.
.
create test/test_helper.rb
create tmp/cache
create tmp/cache/assets
create vendor/assets/javascripts
create vendor/assets/javascripts/.keep
create vendor/assets/stylesheets
create vendor/assets/stylesheets/.keep
run bundle install
.
.
.
Your bundle is complete! Use `bundle show [gemname]` to see where a bundled
gem is installed.
```

Podemos observar las gemas que vienen por defecto en nuestra primera app en el fichero llamado Gemfile. Instalamos el bundle también para nuestra app:

-bundle install

Una vez creada nuestra aplicación podemos iniciar nuestro repositorio local con el comando: git init. Luego podremos ir añadiendo información a nuestro repositorio remoto en Github en el cuál iremos añadiendo código nuevo a medida que avancemos en nuestra app.

Para poder soportar nuestra aplicación podemos utilizar Heroku , mediante a una serie de pasos en el directorio de nuestra app:

-heroku login

-heroku create

-git push heroku master

-heroku open

Para obtener el framework Django nos lo tuvimos que descargar de la página oficial de Django en un fichero del tipo :Django-1.4.5.tar.gz.

<http://django.es/descarga> y luego aplicar el comando de descomprimir el fichero mediante:

-tar xzvf Django-1.4.5.tar.gz.

Luego lo instalamos mediante el comando : cd Django-1.4.3 y sudo python setup.py install. Ahora ya podemos utilizar estos entornos con proyectos sobre ellos.

Si queremos ejecutar Aptana Studio 3 debemos poner el comando:

- aptanastudio3.