

El documento pretende cubrir paso a paso la forma de hacer el deploy de una aplicación de Play2 que usa MongoDB dentro de una instancia de Amazon ec2. En mi caso me basaré en el deploy del Horroscopo

Amazon permite que nos demos de alta en sus servicios básicos gratuitamente durante un año para engancharnos, despues nos terminará cobrando unos diez dolares al mes por los mismos, precio que no considero elevado en absoluto teniendo en cuenta la cantidad de servicios que ofrece.

Al lío:

- Primero nos damos de alta en Amazon, nos hará falta una tarjeta de credito aun cuando el coste sea cero, así ya nos tienen fichados ;-)

<http://aws.amazon.com/es/console/>

- Una vez nos llega el correo de confirmación de la cuenta, nos damos de alta en la consola, pinchamos sobre EC2 -> Instances -> Launch Instance

- Sobre el tipo de instancia a usar, para hacer una cosa rapidita, lo montaremos todo sobre una instancia prefabricada "Quick Start" y a falta de una distro Debian, seleccionaremos la única que se nos ofrece con Ubuntu

- Lo dejamos todo por defecto hasta el paso 5 "Tag Instance" donde asignaremos una par clave->valor con lo siguiente [Name->Horroscopo]

- En el siguiente paso "Configure Security Group" abriremos los puertos necesarios para que nuestro servidor funcione y alguno mas, quien dijo miedo ☺. Debemos abrir para todas las Ips, el puerto 22, 80, 443, 9000, 27017

- Una vez terminamos y pulsamos en "Launch", se nos preguntará por una claves privadas, las cuales deberemos de crear y guardar en un sitio localizado, ya que despues serán las que nos den acceso a la consola

Ya tenemos nuestra maquina virtual preparada, ahora vamos a conectarnos a la shell usando Putty, pero antes hay que configurar alguna cosilla:

- Bajamos el programa “puttygen” si no lo tenemos, le pasamos el archivo de claves que generamos anteriormente “.pem” y guardamos la clave privada en formato “ppk”

- Abrimos “putty” y creamos una nueva conexión con la siguiente configuración:

1. Session->Host Name: ubuntu@<DNS de la instancia ec2> en mi caso (ubuntu@ec2-54-76-126-31.eu-west-1.compute.amazonaws.com) protocolo SSH default
2. Window->Translation: UTF-8 no es imprescindible, pero así eliminamos los caracteres extraños
3. Connection->SSH->Auth: en el campo “Private key ...” le damos la ruta al archivo ppk, generado anteriormente

Ya podemos conectarnos a nuestra máquina, ahora vamos a configurar el entorno para que nuestra aplicación pueda correr libremente

Lo primero, intalar MongoDB y que arranque con el sistema:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen'
| sudo tee /etc/apt/sources.list.d/mongodb.list

sudo apt-get update

sudo apt-get install mongodb-org

sudo service mongod start
```

Ahora instalamos OpenJDK:

```
sudo apt-get install icedtea-7-plugin openjdk-7-jre openjdk-7-jdk
```

Instalamos SBT:

```
wget http://dl.bintray.com/sbt/debian/sbt-0.13.5.deb

sudo dpkg -i sbt-0.13.5.deb

sudo apt-get update

sudo apt-get install sbt
```

Instalamos Git:

```
sudo apt-get install git-core
```

Clonamos desde Github nuestro proyecto original (en mi caso):

```
git clone https://github.com/eliasdefrancisco/horroroscopo.git
```

Cuando haga falta actualizar desde Github, desde la carpeta de nuestro proyecto:

```
git pull
```

Ya tenemos todo listo para probar nuestro proyecto, vamos a ello:

Primero vamos a la carpeta raíz de nuestro proyecto, en mi caso

```
cd ~/horroroscopo/
```

Ahora ejecutamos la aplicación, la primera vez tardara mas porque SBT bajara todas las dependencias

```
sbt
```

Una vez dentro de la consola de SBT, ejecutamos la app en modo DEV:

```
run
```

Si todo fue bien nuestra app ya estará corriendo en el servidor 😊

No aseguramos con nuestro navegador introduciendo como ruta el DNS que nos suministra la consola de Amazon ec2, en nuestro caso

```
http://ec2-54-76-126-31.eu-west-1.compute.amazonaws.com:9000/
```

Todo bien pero....

Como hemos arrancado la aplicación desde la consola en modo DEV, cuando cerramos esta o perdemos la conexión, la aplicación se cae ...

Para arrancar la aplicación en modo PROD, desde la carpeta del proyecto, y que el proceso permanezca activo al cerrar la terminal:

```
sudo nohup sbt play start
```

ctrl+D: Volvemos a la consola sin parar el servidor

ctrl+C: Volvemos a la consola parando el servidor

kill (ps): paramos el servidor desde la consola

nohup: nos devolvera a la consola automáticamente *

Si queremos limpiar caches, y actualizar nuestro proyecto en general:

```
sudo sbt clean compile stage
```

¿Y si nos reinician el servidor?...

Hemos de preparar nuestro sistema para que cargue nuestra app en el arranque del sistema:

```
cd /etc/init.d/  
touch horroroscopo  
chmod 755 horroroscopo
```

Editamos el archivo y metemos el siguiente código para el script de arranque

```
#!/bin/sh  
  
### BEGIN INIT INFO  
  
# Provides:          horroroscopo  
  
# Required-Start:    $local_fs $remote_fs $network  
  
# Required-Stop:     $local_fs  
  
# Default-Start:     1 2 3 4 5  
  
# Default-Stop:  
  
# Short-Description: Arranca el Horroroscopo  
  
# Description: Arranca el Horroroscopo  
  
### END INIT INFO  
  
PATH=/sbin:/bin:/usr/sbin:/usr/bin  
  
. /lib/lsb/init-functions  
. /etc/default/rcS  
  
case "$1" in  
    start)  
        cd /home/ubuntu/horroroscopo/  
        sbt play start  
        ;;  
Esac  
  
exit 0
```

Ejecutamos el siguiente comando para que nuestro script se ejecute en el inicio del sistema

```
sudo update-rc.d horroroscopo enable
```