This document intented to cover how to deploy a Play2 aplication using MongoDB inside an Amazon ec2 instance step by step. In my case I rely on Horroroscopo deploy

Amazon allows us to register their basics services for free for one year to get hooked, then we end up charging about ten dollars a month for the same, price that I don't consider high at all considering the services offered

Come on:

- First we registered in Amazon, we´ll need a credit card even if it is zero cost, and since we have controlled ;-)
http://aws.amazon.com/es/console/

- When we get the confirmation account email, we will log in on the console and click in EC2 -> Instances -> Launch Instance

- We will use a chief recipe for a quick instance configuration, select the one which is offered with Ubuntu

- Leave everything default to step 5 "Tag instance" where we assign a key/value pair with the following [Name->Horroroscopo]

- At next step "Configure Segurity Group" we will open needed ports for our server works and any more, who´s afraid? ☺. We must open this ports for every IPs: 22, 80, 443, 9000, 27017

- When we finished and clicked on "Launch", we will ask for a private keys, which must create and save in a located site, because after will give us the console access

So far we have our virtual machine ready, now we are going to connect in the sell using Putty, but before we must need to configure anything:

- Download the program ”puttygen” if you haven´t, open the key/pair that we generate before “.pem” and save the private key in “ppk” format

- Open “putty” and make a new connection with the follow config:

1. Session->Host Name: ubuntu@<DNS de la instancia ec2> in my case (ubuntu@ec2-54-76-126-31.eu-west-1.compute.amazonaws.com) SSH protocol default
2. Window->Translation: UTF-8 is not essencial, but in my case removes unwanted characters
3. Connection->SSH->Auth: in field “Private key …” we insert the route of the ppk file that generate before

So far we can connect to our machine, now we are going to set up the environment so that our application can run free

First, we will install MongoDB booting with the system

```
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
```

Now install OpenJDK:

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

Now install 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
```

Now install Git:

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

Clone our original proyect from Github (in my case):

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

When need update from Github, from the root directory of the proyect:

```
sudo git pull

(Solo si es necesario)* sudo git reset --hard origin/master
```

So far we have everything ready to test our proyect, come on:

First we go to our proyect root directory, in my case:

```
cd ~/horroroscopo/
```

Now we execute the application, the fisrt time it will take more time because SBT will download all dependencies

```
sbt
```

When we are inside of SBT console, run the app in DEV mode with:

```
run
```

If all went well our application will be running in the server ☺

We ensure this by typing in our browser the DNS route that can be take from the Amazon ec2 console, in my case:

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

All well but…

We was run the application from the console in DEV mode, when we close this or loose the connection, our application shutdown

For run the application in PROD mode and process remains active when close terminal:

```
sudo nohup sbt play start
```

ctrl+D: Go back to the console without stop the server
ctrl+C: Go back to the console stopping the server
kill (ps): stop the server from console
nohup: we will return the console *

If we want clean caches and update the proyect:

```
sudo sbt clean compile stage
```

¿And if the server shutdown?...

We prepare our application for run while booting system:

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

Edit the file and write this code inside

```
#! /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
```

Run this command for our script run on booting system

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