Tarea de introducción a Vagrant

Introducción

Aunque nuestro enfoque será en *Docker*, Vagrant es otra tecnología muy popular que conviene conocer. Es por ello que se nos encargó la tarea de generar un entorno de prueba con una aplicación *Hello, World* bajo **Spring Boot** usando esta tecnología.

Metodología

Instalación de Vagrant y sus requisitos

Vagrant es un software multiplataforma que debe ser instalado en el entorno de trabajo. Además, por debajo hace uso de máquinas virtuales, por lo que también hay que tener el software adecuado para este fin. Una opción preferida y la que usé yo fue VirtualBox.

Una vez instalada, se accede a ella mediante el comando `vagrant`

Figure 1: vgt

VagrantFile

La configuración de la máquina virtual viene dada por el VagrantFile, del que caben destacar las siguiente configuraciones:

- Presentación de puertos: para poder acceder a la máquina fácilmente desde el navegador del *host*

```
  config.vm.network "forwarded_port", guest: 8080, host: 8081, host_ip: "127.0.0.1"
  config.vm.network "forwarded_port", guest: 80, host: 81, host_ip: "127.0.0.1"
```

- Creación de directorios e instalación de paquetes: necesarios para correr nuestra aplicación. **Notar el comentario del proxy**, que es necesario tenerlo en cuenta al instalar los paquetes.

```
config.vm.provision "shell", inline: <<-SHELL
  echo "Configuring my c00l app..."
  echo "Checking if the app directory exists..."
  if [ -d "/home/vagrant/javaApp" ]; then
    echo "Dir already exists!"
  else
    echo "Dir not found! Creating..."
    mkdir /home/vagrant/javaApp
    echo "Do not forget permissions!!"
    sudo chown -R vagrant:vagrant /home/vagrant/javaApp/
    sudo chmod +xrw /home/vagrant/javaApp
   fi
 #export http_proxy=""
 #export https_proxy=""
 apt-get update
 apt-get install -y maven apache2
SHELL
```

- Copia del código y configuración: necesario para poder correr la aplicación al arrancar la máquina

```
config.vm.provision "file", source: "app.jar", destination: "/home/vagrant/javaApp/app.jar"
config.vm.provision "shell", inline: <<-SHELL
  echo "Preparing the application..."
  cd /home/vagrant/javaApp/
  echo "Setting JAVA_HOME"
  export JAVA_HOME="/usr/lib/jvm/default-java"
  echo "Running..."
  java -jar app.jar &
SHELL
```

Configuración y uso de Vagrant

Aunque hay más comandos y conceptos, habiendo ensamblado el fichero anterior, levantar la aplicación es tan sencillo como ejecutar un `vagrant up`, que iniciará y ejecutará la máquina virtual con su aplicación.

Es importante destacar que estos pasos **requieren descargas desde Internet**, por lo que si nos encontramos detrás de un proxy, deberemos tener las variables de entorno debidamente configuradas, de forma similar a:

```
set http_proxy=http://user:password@host:port

set https_proxy=%http_proxy%
```

Con estas variables de entorno seteadas, podemos instalar un plugin de vagrant ya preparado para trabajar con proxy, mediante: `vagrant plugin install vagrant-proxyconf`