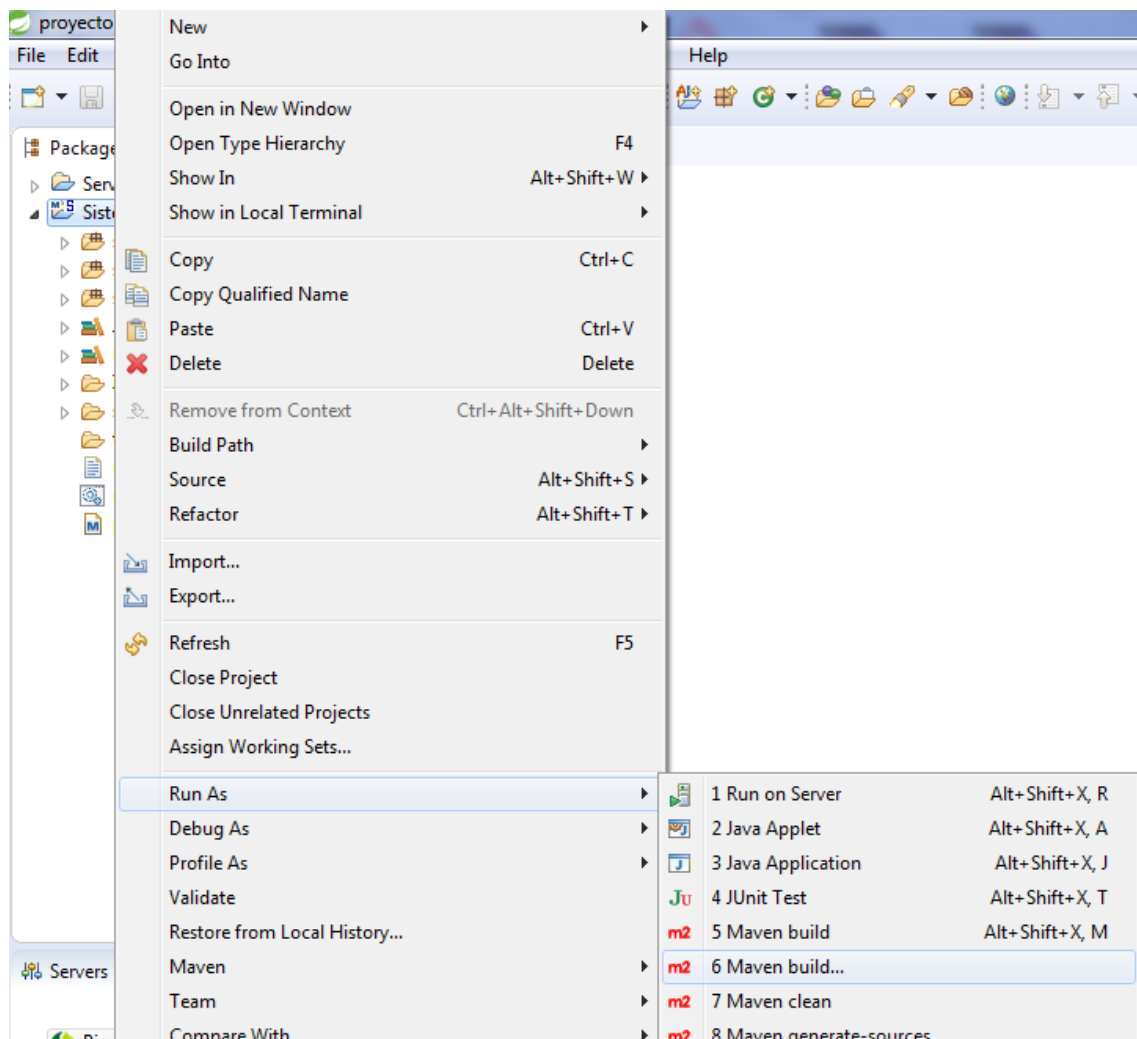
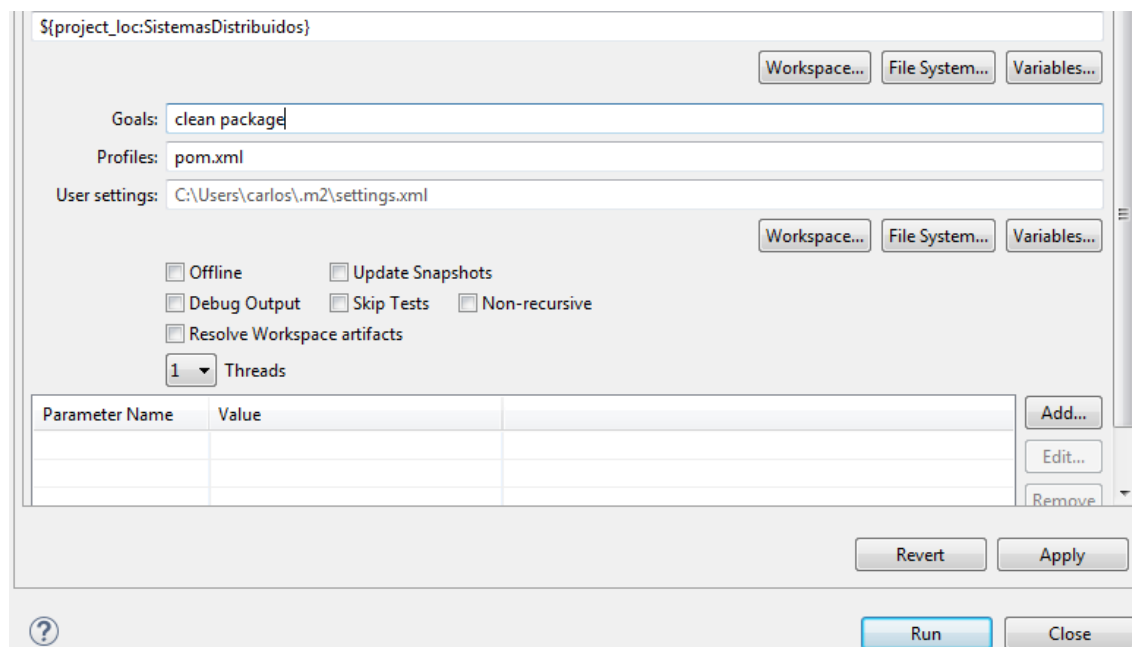


Instrucciones Despliegue Azure

1º- Compilamos el proyecto, para eso entramos en Spring, botón derecho sobre el proyecto que queramos compilar, Run As...Maven Build...



2º- En el apartado Goals introducir clean package y dar Run.



3º- Creación de certificado pem y clave privada, este certificado es necesario para la máquina virtual que instalaremos más tarde.

Para crear dicho certificado y su respectiva clave privada, instalaremos Git para Windows, concretamente Git Bash, lo abrimos e introducimos la siguiente instrucción.

openssl.exe req -x509 -nodes -days 365 -newkey rsa: 2048 -keyout myPrivateKey.key -out myCert.pem. (Los campos en rojo son a nuestra elección. podemos poner el nombre que nosotros deseemos. Siempre que terminen en .key y .pem respectivamente)

```
carlos@carlos-PC MINGW32 ~
$ openssl.exe req -x509 -nodes -days 365 -newkey rsa:2048 -keyout myPrivateKey.key -out myCert.pem
Generating a 2048 bit RSA private key
.....+++
.+++
writing new private key to 'myPrivateKey.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:URJC
Organizational Unit Name (eg, section) []:Code
Common Name (e.g. server FQDN or YOUR name) []:Rosa Roman
Email Address []:rm.roman@alumnos.urjc.es
```

Como observamos deberemos introducir algunos datos para completar el certificado.

4º- Como medida de seguridad, establecemos los permisos para la clave privada con la siguiente instrucción.

chmod 0600 myPrivateKey.key

```
carlos@carlos-PC MINGW32 ~
$ chmod 0600 myPrivateKey.key

carlos@carlos-PC MINGW32 ~
$
```

5º- Creación de una clave privada para PuTTY. Para poder usar PuTTY como cliente ssh, debemos hacer una clave privada especial para PuTTY, para ello usamos GitBash y convertimos la clave privada myPrivateKey.key en una rsa myPrivateKey_rsa, para ello usamos el siguiente comando.

openssl rsa -in ./myPrivateKey.key -out myPrivateKey_rsa

```
carlos@carlos-PC MINGW32 ~
$ openssl.exe rsa -pubout -in myPrivateKey.key -out myPublicKey.rsa
writing RSA key

carlos@carlos-PC MINGW32 ~
$ |
```

6º- Una vez creada nuestra clave privada rsa, establecemos los permisos como hemos realizado anteriormente con la clave myPrivateKey.key.

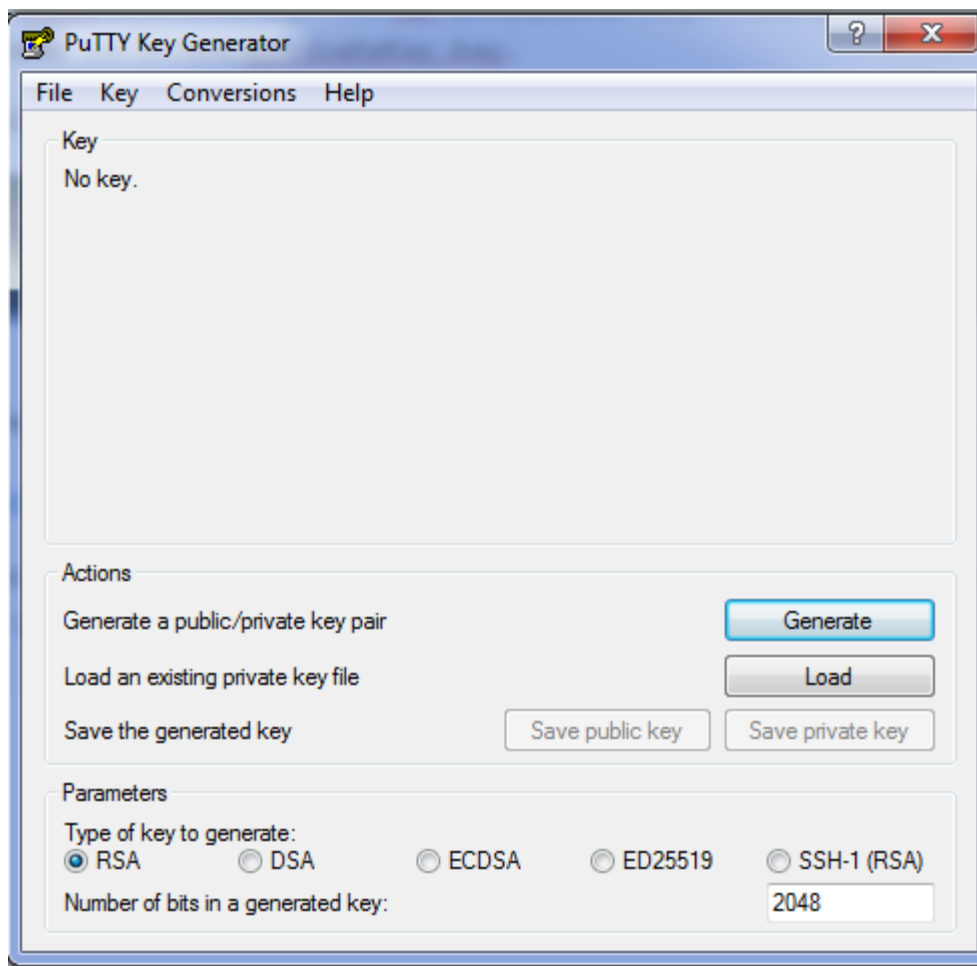
```
chmod 0600 myPrivateKey_rsa
```

```
carlos@carlos-PC MINGW32 ~  
$ chmod 0600 myPrivateKey_rsa  
carlos@carlos-PC MINGW32 ~  
$ |
```

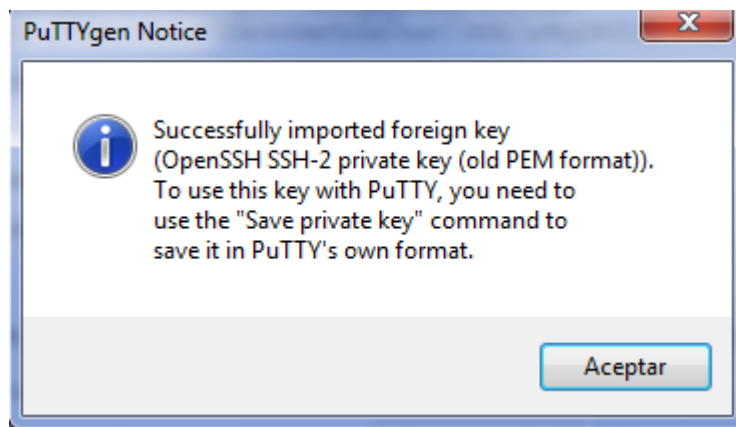
7º- El siguiente paso es descargar PuTTY, podemos hacerlo desde el siguiente enlace.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

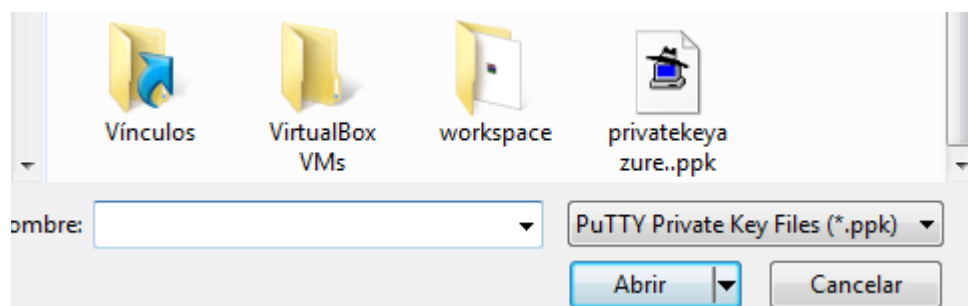
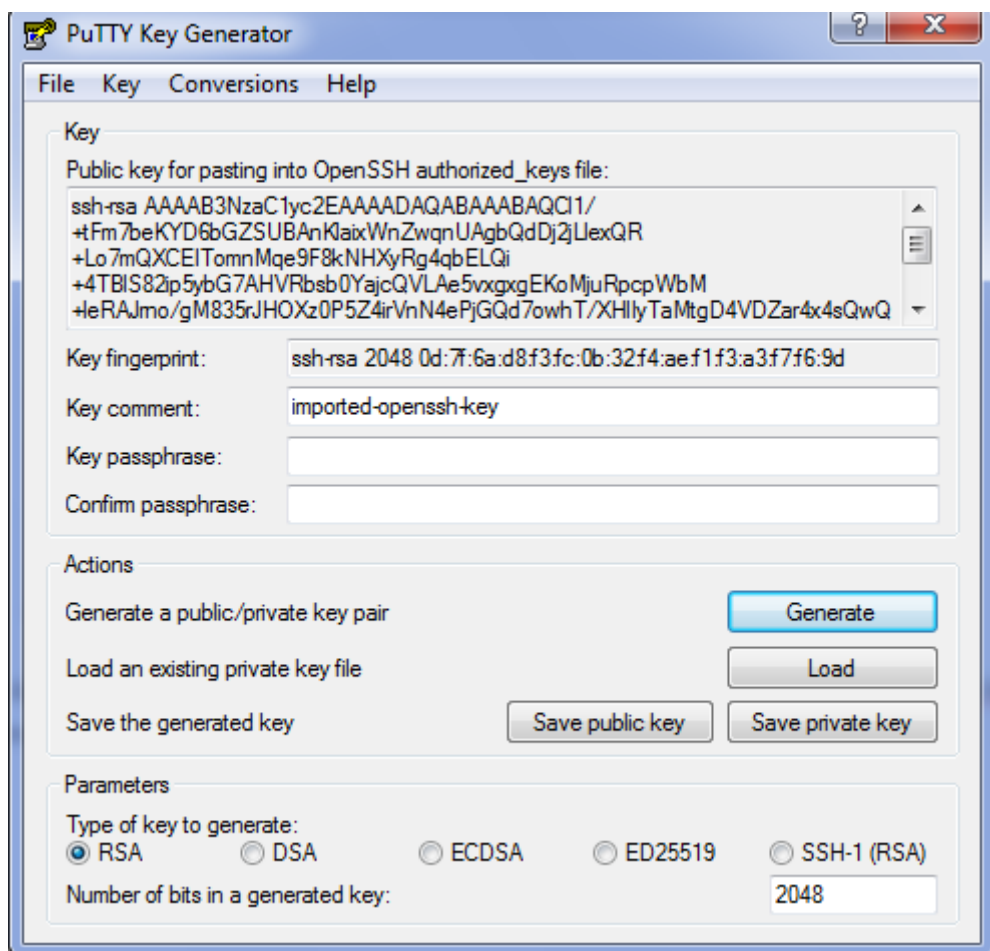
8º- Una vez descargado, abrimos PuTTYgen para generar la clave privada que deberemos introducir más adelante para usar PuTTY.



9º- Dentro de esta pantalla, hay que seguir los siguientes pasos, File-->Load Private Key->Buscamos el directorio del archivo->Seleccionamos en la pantalla All File(*.*)->Cargamos el archivo rsa anteriormente generado. Saldrá la siguiente imagen y le damos aceptar.



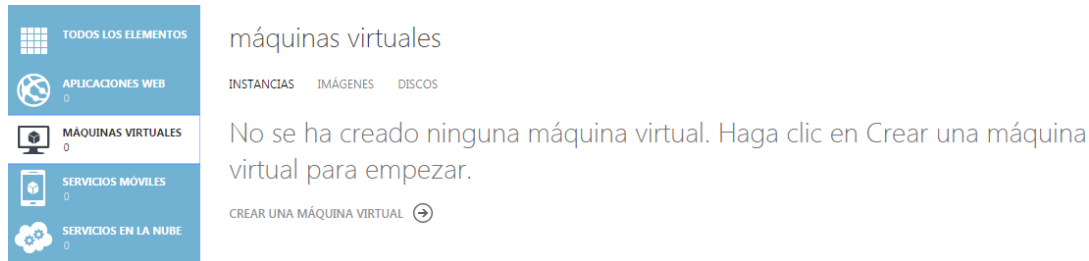
10º- En la siguiente pantalla que nos aparece, tenemos que seleccionar Save Private key, y obtendremos la clave privada para poder usar PuTTY.



11º- Vamos a la página oficial de Azure para crear nuestra Máquina Virtual.

<https://manage.windowsazure.com>

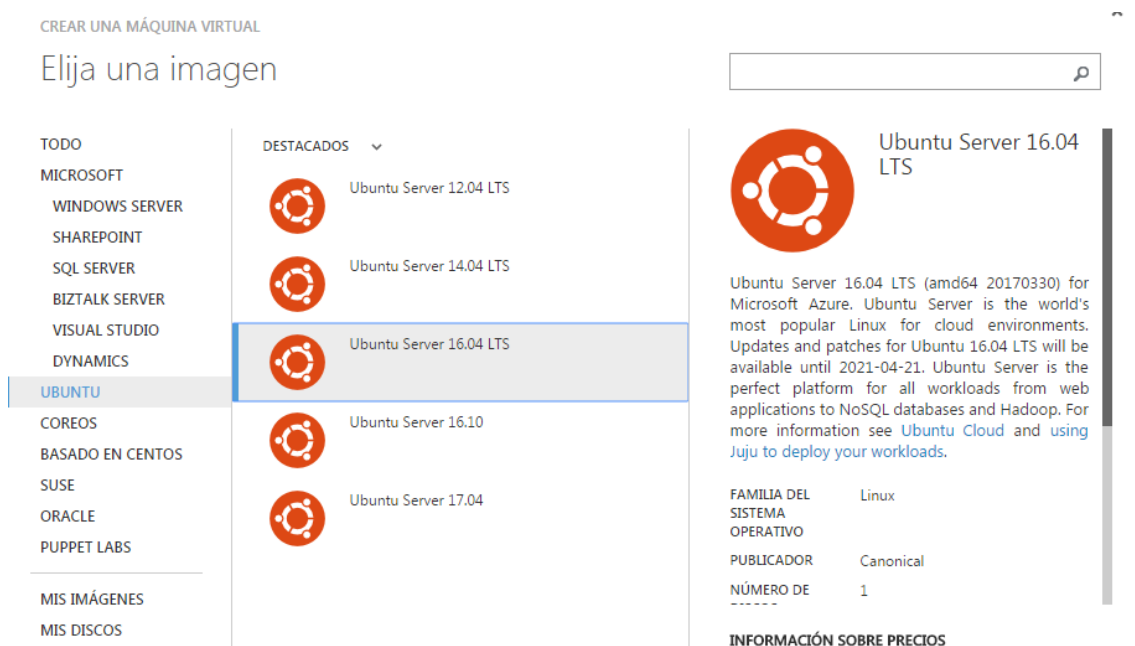
12º- Una vez allí, metemos nuestro correo, (debe de ser una cuenta LiveID), y nuestra contraseña, y empezamos con el asistente de instalación.



13º- Navegamos por el menú y elegimos una máquina de la galería.



14º- Elegimos Ubuntu Server 16.04 LTS



15º- Introducimos el nombre de la máquina y sobre todo el certificado .pem, que nos valdrá para acceder a la máquina.

Configuración de la máquina virtual

30/03/2017

NOMBRE DE LA MÁQUINA VIRTUAL ?
peluqueria

NIVEL
BÁSICO ESTÁNDAR


TAMAÑO ?
A1 (1 núcleo, 1,75 GB de memoria)

NUEVO NOMBRE DE USUARIO
azureuser

AUTENTICACIÓN ?
☒ CARGAR CLAVE DE SSH COMPATIBLE PARA AUTENTICACIÓN

CERTIFICADO
myCert.pem

☐ PROPORCIONAR UNA CONTRASEÑA

 Ubuntu Server 16.04 LTS

Ubuntu Server 16.04 LTS (amd64 20170330) for Microsoft Azure. Ubuntu Server is the world's most popular Linux for cloud environments. Updates and patches for Ubuntu 16.04 LTS will be available until 2021-04-21. Ubuntu Server is the perfect platform for all workloads from web applications to NoSQL databases and Hadoop. For more information see [Ubuntu Cloud](#) and [using Juju to deploy your workloads](#).

FAMILIA DEL SISTEMA OPERATIVO
Linux

PUBLICADOR
Canonical

INFORMACIÓN SOBRE PRECIOS
Los precios varían en función de la suscripción que seleccione para aprovisionar la máquina virtual.

16º- Las siguiente paginas son de configuración y de validación, una vez terminemos el proceso, ya tendremos nuestra máquina virtual, que sería de este modo.

TODOS LOS ELEMENTOS

APLICACIONES WEB

MÁQUINAS VIRTUALES

SERVICIOS MÓVILES

SERVICIOS EN LA NUBE

máquinas virtuales

INSTANCIAS IMÁGENES DISCOS

NOMBRE	ESTADO	SUSCRIPCIÓN	UBICACIÓN	NOMBRE DE DNS
peluqueria	✓ En ejecución	Pase para Azure	Norte de Europa	peluqueria.cloudapp.net

17º- Una vez tengamos la máquina virtual, subiremos el .Jar generado anteriormente, para ello Descargamos WINSCP portable, podemos hacerlo en el siguiente enlace.

<https://winscp.net/download/WinSCP-5.9.4-Portable.zip>

18º- Una vez descargado, lo ejecutamos, introducimos los datos del Host y de damos a advanced.

New Site

Session

File protocol:
SFTP

Host name:
peluqueria.cloudapp.net

Port number:
22

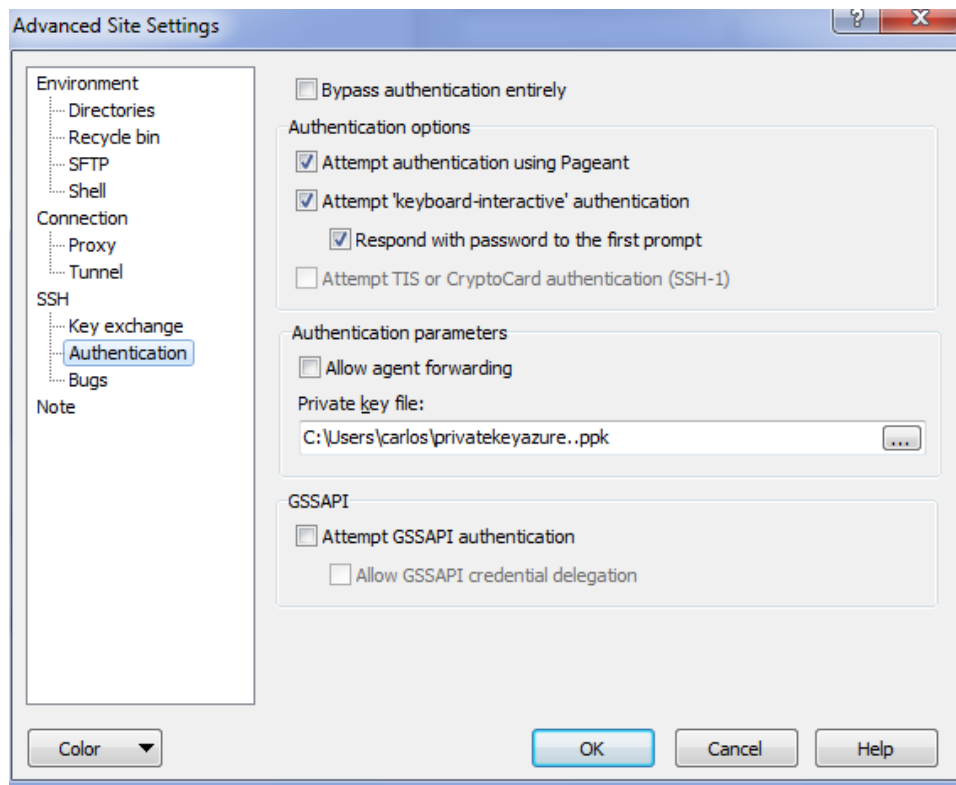
User name:

Password:

Save

Advanced...

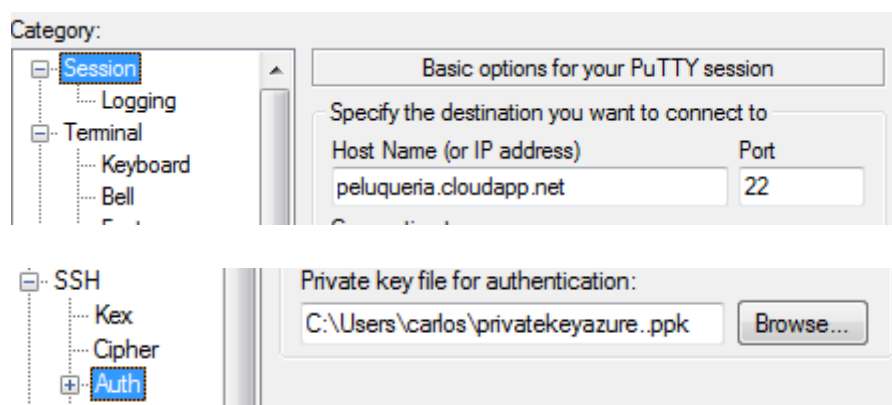
19º- En advanced debemos ir a SSH Authentication e introducir la clase generada anteriormente para PuTTY.



20º- Le damos a login, introducimos el usuario de Azure y ya podremos subir el .Jar a nuestra máquina virtual.

C:\Users\carlos\Desktop\proyecto spring\SistemasDistribuidos\target					/home/azureuser				
Name	Size	Type	Changed		Name	Size	Changed	Rights	Owner
.		Parent directory	24/04/2017		..		23/04/2017 20:58:20	rw-r-xr-x	root
classes		Carpeta de archivos	24/04/2017		nohup.out	180 KB	24/04/2017 19:20:14	rw-r--r--	azureu...
generated-sources		Carpeta de archivos	24/04/2017		Peluqueria-0.0.1-SNA...	32.475 KB	24/04/2017 17:13:55	rw-rw-r--	azureu...
generated-test-sources		Carpeta de archivos	24/04/2017						
maven-archiver		Carpeta de archivos	24/04/2017						
maven-status		Carpeta de archivos	24/04/2017						
surefire-reports		Carpeta de archivos	24/04/2017						
test-classes		Carpeta de archivos	24/04/2017						
Peluqueria-0.0.1-SNAPSHOT.jar	32.475 KB	Executable Jar File	24/04/2017						
Peluqueria-0.0.1-SNAPSHOT.jar.original	29 KB	Archivo ORIGINAL	24/04/2017						

21º- Una vez realizado todo este proceso nos disponemos instalar lo necesario para que funcione nuestra aplicación, lo primero que debemos hacer es abrir PuTTY. Exe, allí, debemos introducir el nombre del Host y la clave privada generada para PuTTY



22º- Introducimos nuestro usuario de Azure y nos disponemos a introducir los comandos necesarios.

23º- Lo primero es instalar el Open JDK 8 JRE, introduciendo los siguientes comandos.

```
azureuser@peluqueria:~$ sudo add-apt-repository ppa:openjdk-r/ppa
```

```
azureuser@peluqueria:~$ sudo apt-get update
```

```
azureuser@peluqueria:~$ sudo apt-get install openjdk-8-jre
```

24º- Después instalaremos MySQL, también con los siguientes comandos.

```
azureuser@peluqueria:~$ sudo apt-get update
```

```
azureuser@peluqueria:~$ sudo apt-get install -y mysql-server
```

25º- Configuramos la BBDD accediendo como root.

```
azureuser@peluqueria:~$ mysql -u root -p
```

26º- Una vez dentro creamos nuestra base de datos, en este caso se llama pelu.

```
mysql> CREATE DATABASE pelu;
```

27º- Una vez terminado el proceso, comprobamos que efectivamente se ha creado.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| pelu |
| performance_schema |
| sys |
+-----+
```

28º- Salimos de mysql.

```
mysql> exit
Bye
```

29º- Por último ejecutamos nuestra aplicación con el comando nohup delante para siga corriendo aunque cerremos ssh.

```
azureuser@peluqueria:~$ nohup java -jar Peluqueria-0.0.1-SNAPSHOT.jar &
[1] 19226
```

30º- Una vez hemos realizado todos estos pasos, podemos acceder a la aplicación desde:

<https://peluqueria.cloudapp.net>.