Cómo integrar Jupyter Hub

Pipoyan-Paronyan, Arman

arman.pipoyan26@gmail.com
Bellaterra, 2022

1. Creación de usuarios en Linux

Se necesita que cada usuario que este registrado en la página web tenga un usuario en la máquina y que esté dentro de un grupo él solo, así solo él será capaz de modificar sus archivos. Para ello tendremos que modificar los siguientes ficheros:

- *Controller/register.php*: En la línea 44 (antes de redireccionar al usuario) hacer que se ejecute, usando la función *shell_exec* [1], el comando *useradd* para crear al usuario nuevo en el SO. El nombre será el correo que está en la variable *email* y se deberá generar una contraseña aleatoria [2] que se guardará en la base de datos usando funciones como *registerProfessor* o *registerProfessor* que en vez de un 'Insert' hagan un 'Update' en la base de datos del usuario con la nueva contraseña. Para guardarlo en la base de datos primero se deberá crear en las tablas 'student' y 'professor' este campo, que tendrá como valor por defecto NULL.

- *Controller/editor.php*: En la línea 100, después de que se termine de crear la carpeta del usuario, de la asignatura i del ejercicio al que se ha accedido, deberemos ejecutar el comando *chown* para hacer que todas las carpetas del directorio del usuario sean propiedad del usuario www-data y del grupo al que pertenece el usuario.

Una vez hecho esto, el usuario debería poder modificar solo los ficheros de su carpeta de soluciones tanto si no está usando Juptyer, porque se controla que no use librerías que le posibiliten ejecutar comandos, como si lo está usando, porque se lo impedirá el propio sistema operativo.

2. JupyterHub

Para hacer que cada usuario tenga su propia instancia de Jupyter Notebook se deberá usar JupyetHub dejando la autenticación que lleva puesta, que permite solo acceder a usuarios que tengan una cuenta en la máquina. Además, se tendrá que crear una clave ssl con tal de poder tener el servidor escuchando en el puerto 443. Esto se puede hacer con el comando 'openssl req -sha256 -new -key private.pem -out csr.pem' y 'openssl x509 -req -sha256 -days 365 -in csr.pem -signkey private.pem -out certificate.pem'.

Se tendrá que crear un fichero de configuración usando el comando *jupyterhub –generate-config* y modificar los siguientes elementos:

- *c.JupyterHub.bind_url = 'https://0.0.0.0:443'*: Esto hará que se acepten peticiones desde cualquier IP del puerto 443.

- *c.Spawner.default_url = '/tree/'*: Esto hará que la página a la que se redireccione al usuario al acceder a su servidor sea '/tree/', que es donde están listados todos sus ficheros.

- *c.Spawner.notebook_dir = '/var/www/tfg/app/solucions/username'*: Esto hará que en el directorio '/tree/' se muestren solo los archivos de la carpeta de soluciones del usuario, aunque esto no evita que el usuario pueda navegar a otras rutas.

Además, se deberá modificar en el fichero *View/js/editor.js*.

- Para construir la URL del iframe es necesario el e-mail del usuario, que se puede obtener haciendo *let email = fileName.split('/').slice(-4)[0];*.

- Teniendo el e-mail, se debe modificar la URL de la línea 287 a la siguiente: *https://158.109.64.206/user/$email/tree/$fileLocation*.

Con estos cambios se debería ser capaces de visualizar y editar los ficheros '.ipynb' mientras esté JupyterHub iniciado con este comando: 'jupyterhub -f /path/to/custom/jupyterhub_config.py –ip 127.0.0.1 –port 443 –ssl-key /path/to/private.pem –ssl-cert /path/to/certificate.pem'

3. Ficheros innecesarios

Con estos cambios los ficheros de la carpeta *jupyter/*, *Controller/removeJupyterDocker.php*, *Model/dockerUtils.php* y las líneas de la 45 a la 54 del fichero *Controller/editor.php* se podrían borrar.

Referencias

[1] "Method to execute operating system commands." https://www.php.net/manual/es/function.shell-exec.php.

[2] "Random string generator." https://stackoverflow.com/a/6101969.