

Python

Utilizar Python en TeideHPC

Una vez conectados a los nodos de login, disponemos de una versión de Python por defecto, que es la versión 2.6.6. Para ver las versiones de Python disponibles, utilizamos la herramienta de *modules*:

```
module ava python

----- /opt/envhpc/modulefiles/.rhel6
-----
python/2.7.18/gcc python/3.5.4/gcc python/3.7.9/gcc python/3.8.11/gcc
```

Para cargar alguna de las versiones es posible que tengamos que cargar algún módulo previamente:

```
module load python/3.8.11/gcc
python/3.8.11/gcc(10):ERROR:151: Module 'python/3.8.11/gcc' depends on one of the
module(s) 'openssl/1.1.1k/gcc'
python/3.8.11/gcc(10):ERROR:102: Tcl command execution failed: prereq openssl/1.1.1k/gcc
```

Por tanto, cargamos los módulos necesarios, en el orden correspondiente:

```
module load openssl/1.1.1k/gcc python/3.8.11/gcc

module list
Currently Loaded Modulefiles:
  1) openssl/1.1.1k/gcc  2) python/3.8.11/gcc
```

Ahora podremos utilizar una versión específica de Python y utilizar entornos, donde instalar paquetes de Python de manera aislada.

Instalar paquetes de Python en /data

Por defecto, Python instalará los paquetes en el `/home` del usuario. Este almacenamiento está limitado en tamaño, así que deberíamos configurar los entornos e instalar el software en la partición de `/data`. Para hacer esto, tenemos que hacer lo siguiente:

```
pip3 install --target=$HOME/data/tutu --install-option="--install-scripts=$HOME/data/foo"
Package
```

Con la opción `--target=DIR` le indicamos a pip donde se tienen que instalar tanto el paquete solicitado, como todas sus dependencias. Por otro lado, la opción `--install-option="--install-scripts=DIR`, indica donde irán instalados los binarios, en caso de utilizar el paquete de python desde la línea de comandos directamente.

Luego, tenemos que añadir a nuestro PATH la nueva ruta para utilizar los binarios directamente:

```
export PYTHONPATH="$HOME/data/tutu"
```

Para hacerlo permanente, podemos añadir esa línea al final de nuestro `~/.bashrc`.