

Environment modules TCL

DEPRECATED

Debido a la actualización del sistema operativo de los nodos de cómputo esta página de documentación quedará obsoleta. La nueva herramienta a usar será

Lmod

Puesto que es usted está en un entorno compartido por cientos de usuarios, la forma de proveer multiples aplicaciones en sus múltiples versiones a multitud de usuarios en un entorno de computo distribuido no es la misma que en un ordenador personal o un servidor propio.

La herramienta para la gestión de estos entornos se llama **Environment Modules**. Puede encontrar una explicación sobre Environment modules en el siguiente enlace:

- <http://modules.sourceforge.net>
- <https://modules.readthedocs.io/en/stable/module.html>

Básicamente **Environment modules** son un *conjunto de scripts* que nos permiten modificar fácilmente el entorno de usuario (la shell) haciendo disponibles al usuario sólo las aplicaciones que queramos usar. Por tanto, será la herramienta a usar para **cargar/descargar aplicaciones** una vez se inicie sesión en los nodos de login.

¿Cómo usar environment modules?

En apartados posteriores verá cómo se usan junto al gestor de colas de trabajos **Slurm** pero para familiarizarse puede **iniciar sesión** en un nodo de login y ejecutar los siguientes comando y ejemplos para entender su funcionamiento:

```
$module avail (o av)          -- muestra los módulos que tenemos disponibles
```

```
$module load modulename (o add) -- carga un módulo/aplicación concreto
```

```
$module unload modulename (o rm) -- eliminar un módulo cargado
```

```
$module purge                -- elimina todos los módulos cargados
```

```
$module list -- lista los módulos cargados por el usuario
```

```
$module show modulename -- muestra la información de módulo
```

```
$module whatis modulename -- muestra la información del modulefile
```

```
$module update modulename -- recarga todos los módulos cargados
```

```
$module help -- muestra información de ayuda del módulo
```

Ejemplos

```
$ which gcc -> /usr/bin/gcc  
$ module load gcc/10.2.0  
$ which gcc -> /opt/envhpc/utils/rhel6/gcc/10.2.0/bin/gcc
```

Puede ver cómo el ejecutable de la aplicación que se está usando antes y después de cargar el módulo.

```
$ module av 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
```

```
$ which python -> /usr/bin/python  
$ which python3 -> /usr/bin/which: no python3 in (/usr/local/bin:...)   
  
$ module load python/3.7.9/gcc -> Requisito cargar openssl/1.1.1k/gcc  
  
python/3.7.9/gcc(10):ERROR:151: Module 'python/3.7.9/gcc' depends on one of the  
module(s) 'openssl/1.1.1k/gcc'.  
python/3.7.9/gcc(10):ERROR:102: Tcl command execution failed: prereq openssl/1.1.1k/gcc.  
  
$ module load openssl/1.1.1k/gcc python/3.7.9/gcc  
$ which python3 -> /opt/envhpc/utils/rhel6/python/3.7.9/gcc-10.2.0/bin/python3
```

En este caso, como primer paso, busca alguna versión de python disponible en el sistema.

Seguidamente, observará que comando *python3* no está disponible.

Al cargar el módulo *python/3.7.9/gcc* se pide como requisito para su correcto funcionamiento precargue el módulo *openssl/1.1.1k/gcc*.

Una vez cargados estos dos módulos, puede observar cómo la versión 3.7.9 de python está disponible.

¿Cómo usar módulos en un script?