

Jupyter Notebook

Ejecución remota de Jupyter Notebook con Slurm

Como sabemos, Jupyter Notebook se pueden iniciar localmente y acceder a los sistemas de archivos locales, o se pueden iniciar en una máquina remota y que proporcione acceso a los archivos de un usuario en el sistema remoto. En este último caso, Jupyter se inicia a través de un proceso que crea una URL única que se compone del nombre de host más un puerto disponible (elegido por la aplicación jupyter) más un token único. El usuario obtiene esta URL y la ingresa en un navegador web local, donde el servidor de Notebooks está disponible siempre que el proceso en la máquina remota esté activo y en ejecución. Hay que tener en cuenta que de forma predeterminada, estos servidor de Jupyter no es seguro y exponen potencialmente los archivos locales de los usuarios a usuarios no deseados.

Por ello debes tener en cuenta estos conceptos:

■ No ejecutar Jupyter en los nodos de login

Los nodos de login del clúster es un recurso compartido por muchos usuarios. La ejecución de Jupyter en uno de estos nodos puede afectar negativamente a otros usuarios. **Utilice uno de los enfoques descritos en esta página para realizar su trabajo.**

■ Internet no está disponible en los nodos de cómputo

Las sesiones de Jupyter deben ejecutarse en los nodos de cómputo los cuales no tienen acceso a Internet. Esto significa que no podrá descargar archivos, clonar un repositorio de GitHub, instalar paquetes, etc. **Deberá realizar estas operaciones en los nodos de login.** Cualquier archivo que descargue mientras está en el nodo de login estará disponible inmediatamente en los nodos de cómputo durante la sesión.

■ No puede acceder directamente a los nodos de cómputo

Por motivos de seguridad, no se puede acceder directamente a los nodos de cálculo. Sin embargo, se puede acceder a ellos mientras ejecuta un trabajo desde cualquier nodo de login.

■ Esta implementación no es el servidor multiusuario que andas buscando

Este documento describe cómo puede ejecutar un servidor público con un solo usuario. Esto solo debe hacerlo alguien que quiera acceder de forma remota a su cuenta personal. Aun así, hacer esto requiere una comprensión profunda de las limitaciones de las configuraciones y las implicaciones de seguridad. Si permite que varios usuarios accedan a un servidor portátil como se describe en este documento, sus comandos pueden colisionar, aplastarse y sobrescribirse entre sí.

Si quieres un servidor multiusuario, la solución oficial es JupyterHub. Para usar JupyterHub, necesita un servidor Unix (generalmente Linux) que se ejecute en algún lugar al que puedan acceder sus usuarios en una red. Esto puede ejecutarse a través de Internet público, pero hacerlo presenta problemas de seguridad adicionales.

Crear un entorno conda

En primer lugar tenemos que crear un entorno conda. Puede ver estos pasos [aquí] (how_to_conda.md).

Recordar:

- Use nodos de inicio de sesión para crear un entorno python/conda. Sólo estos tienen acceso a internet.
- **Instalar paquetes python o conda en /data**

```
$ module spider miniconda
$ module load Miniconda3/4.9.2
$ conda create --name jupyter-env [python=3.9.15] or
$ conda create --prefix /home/user/data/allenvironments/my_environment [python=3.9.15]
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done
....
# To activate this environment, use
# $ conda activate jupyter-env
# To deactivate an active environment, use
# $ conda deactivate
```

- Instalar los paquetes necesarios para Jupyter Notebooks

```
conda activate jupyter-env
conda install jupyter ipykernel matplotlib ipywidgets ipympl --channel conda-forge -y
```

Ejecutar Jupyter Notebook en un nodo de cómputo mediante *salloc*

Las tareas más grandes se pueden ejecutar en uno de los nodos de cómputo solicitando una sesión interactiva usando *salloc*.

Lo primero es lo primero: inicia una sesión de **screen** (o *tmux* si lo prefieres). Si busca tener un programa ejecutándose por más tiempo del que quiero mantener abierta una ventana de terminal, *screen* o *tmux* son excelentes opciones, ya que evitan que su sesión se agote en máquinas remotas.

```
screen -S jupyter
salloc --nodes=1 --partition batch --ntasks=1 --mem=20G --time=12:00:00
# or salloc -N 1 -p batch
# or srun -p batch --pty bash

conda activate jupyter-env
```

Ahora debería ver que el indicador de su terminal ha cambiado a algo como lo siguiente, lo que indica que ha iniciado sesión interactiva un nodo de cómputo y está trabajando dentro del entorno *conda-env*:

```
(jupyter-env) yourUser@nodeXXXX-X
```

Una vez que se ha asignado un nodo de cómputo, inicia Jupyter. En el nodo de cómputo ejecuta:

```
jupyter-notebook --no-browser --port=8889 --ip=127.0.0.1
```

Jupyter Notebook se inician a través de un proceso que crea una URL única que se compone de ip y puerto 8889 más un token de un solo uso. El usuario obtiene esta URL y la ingresa en un navegador web local, donde el servidor está disponible siempre que el proceso en la máquina remota esté activo y en ejecución.

```
[I 13:22:01.198 NotebookApp] Writing notebook server cookie secret to /home/youruser/.local/share/jupyter/runtime/notebook_cookie_secret
```

```
[I 13:22:12.448 NotebookApp] Serving notebooks from local directory: /home/youruser
```

```
[I 13:22:12.448 NotebookApp] Jupyter Notebook 6.5.2 is running at:
```

```
[I 13:22:12.448 NotebookApp] http://127.0.0.1:8889/?
token=36229e08e0944c8d1b4df0174e23a7ee11e278ccbed5f967
```

```
[I 13:22:12.448 NotebookApp] or http://127.0.0.1:8889/?
token=36229e08e0944c8d1b4df0174e23a7ee11e278ccbed5f967
```

[I 13:22:12.448 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation). [C 13:22:12.568 NotebookApp]

To access the notebook, open this file in a browser:

file:///home/youruser/.local/share/jupyter/runtime/nbserver-4762-open.html

Or copy and paste one of these URLs:

http://127.0.0.1:8889/?token=36229e08e0944c8d1b4df0174e23a7ee11e278ccbed5f967

or http://127.0.0.1:8889/?token=36229e08e0944c8d1b4df0174e23a7ee11e278ccbed5f967

Tip

Tenga en cuenta que de forma predeterminada Jupyter Notebook abre automáticamente un navegador, pero *no podemos hacerlo en un nodo de cómputo directamente por seguridad*, por lo que omitimos esa función con el indicador *--no-browser*.

Tip

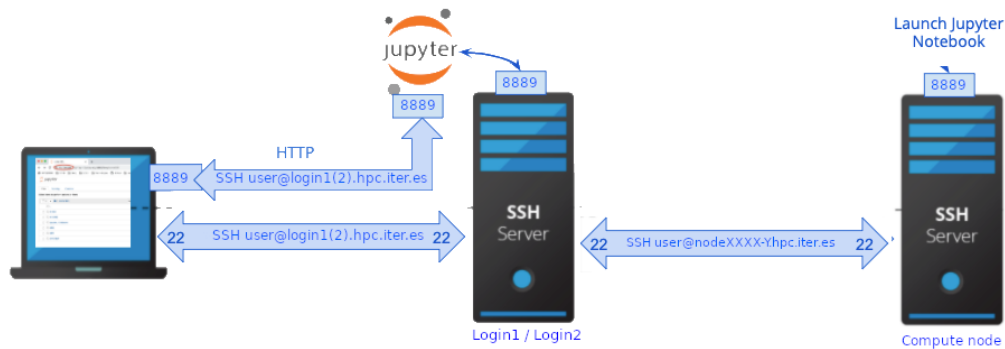
Tenga en cuenta que seleccionamos el puerto Linux 8889 para conectarse al servidor de Notebook. Si no especifica el puerto, se establecerá de forma predeterminada en el puerto 8888, pero a veces este puerto puede estar ya en uso en la máquina remota o en la local (es decir, su PC). Si el puerto que seleccionó no está disponible, recibirá un mensaje de error, en cuyo caso debe elegir otro. Es mejor seleccionar un puerto mayor al 1024. Considere comenzar con 8888 e incrementarlo en 1 si falla, por ejemplo, intente 8888, 8889, 8890 y así sucesivamente. Si está ejecutando en un puerto diferente, sustituya su número de puerto por 8889.

Por defecto, Jupyter Notebooks no es seguro y potencialmente exponen sus archivos a usuarios no deseados.

Ejecutar Notebook de la forma habitual utiliza conexiones HTTP inseguras. En esta guía, **presentamos una guía para ejecutarlo de forma más segura.**

Crear un tunnel SSH

Por razones de seguridad, recomendamos usar túneles ssh para conectarse de forma segura al servidor de Jupyter Notebook. Con esto creará una conexión ssh entre su host local y el puerto del servidor de Notebook ejecutándose en el nodo interactivo remoto. Cuando conecte su navegador al servicio de notebook, este canalizará todas las comunicaciones a través de la conexión SSH, que es segura y encriptada.



Inicie una segunda sesión de terminal en su máquina local, conéctese a cualquier nodo de login y configure el túnel SSH de la siguiente manera:

```
ssh yourUser@loginX.hpc.iter.es
```

En el nodo de login escribir:

```
ssh -N -L localhost:8889:localhost:8889 yourUser@nodeXXXX-Y.hpc.iter.es
```

Podrias ver un mensaje de este tipo:

```
The authenticity of host 'node1710-1.hpc.iter.es (10.0.17.37)' can't be established.
ECDSA key fingerprint is SHA256:LqRpSE90hft08tO47V2nx7jbIsUOX42SeKAgEVBPoDo.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added 'node1710-1.hpc.iter.es' (ECDSA) to the list of known hosts.
```

Inicia un tercera consola en tu ordenador personal y establece el tunnel hacia el nodo de login de la siguiente manera.

```
ssh -N -L localhost:8889:localhost:8889 yourUser@loginX.hpc.iter.es
```

-N No ejecuta un comando remoto. Útil sólo para reenviar puertos.

-L Especifica que el puerto dado en el host local (cliente) se reenviará al puerto dado host y puerto en el lado remoto.

Puedes usar los DNS o las IPs de los nodos

Sólo hay que cambiar *nodeXXXX-X.hpc.iter.es* por la IP del nodo.

Finalmente, copie y pegue la dirección en su navegador favorito y **reemplace la parte “?token=x” de la URL con su token**

http://127.0.0.1:8889/?token=36229e08e0944c8d1b4df0174e23a7ee11e278ccbed5f967

Recomendamos leer sa sección **Notas importantes** below.



Files Running Clusters

To import a notebook, drag the file onto the listing below or [click here](#). New ▾ ↻

/ [examples](#)

<input type="checkbox"/>	..
<input type="checkbox"/>	Builtin Extensions
<input type="checkbox"/>	Customization
<input type="checkbox"/>	Embedding
<input type="checkbox"/>	IPython Kernel
<input type="checkbox"/>	Interactive Widgets
<input type="checkbox"/>	Notebook
<input type="checkbox"/>	Parallel Computing
<input type="checkbox"/>	images
<input type="checkbox"/>	utils
<input type="checkbox"/>	Index.ipynb

Running Jupyter Notebook on a Compute Node via *sbatch*

La segunda forma de ejecutar Jupyter en el clúster es enviando un trabajo a mediante el commando *sbatch*.

Para hacer esto, necesitamos un script de envío como el siguiente que podemos llamar **jupyter.sh**:

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=20G
#SBATCH --partition=batch
#SBATCH --constrains=sandy
#SBATCH --time=01:00:00
#SBATCH --job-name=jupyter-notebook
#SBATCH --output=jupyter-notebook-%j.out
#SBATCH --error=jupyter-notebook-%j.err
#####
# Enable modules profile

# Enable conda on bash console
eval "$(conda shell.bash hook)"
# load modules or conda environments here
```

```

module load Miniconda3/4.9.2
conda activate jupyter-env

# get tunneling info
XDG_RUNTIME_DIR=""
node=$(hostname -s)
user=$(whoami)
cluster="teide-hpc"
port=8889

# print tunneling instructions jupyter-log
echo -e "
Command to create ssh tunnel from login node to compute node:
ssh -N -L localhost:${port}:localhost:${port} ${user}@nodeXXXX-Y.hpc.iter.es

Command to create ssh tunnel from your local machine to any login node:
ssh -N -L localhost:${port}:localhost:${port} ${user}@login1(2).hpc.iter.es

Use a Browser on your local machine to go to:
localhost:${port} (prefix w/ https:// if using password)
"

# Run Jupyter
jupyter-notebook --no-browser --port=${port} --ip=127.0.0.1

```

Este trabajo inicia Jupyter en el nodo de cómputo asignado y podemos acceder a él a través de un túnel ssh como lo hicimos en la sección anterior.

Enviamos el trabajo a la cola:

```

sbatch jupyter-notebook.sh

```

Una vez que se ejecuta el trabajo, se creará un log llamado *jupyter-notebook-.log*. Este log contiene información sobre cómo conectarse a Jupyter y el token necesario.

Para conectarse a Jupyter que se ejecuta en el nodo de cómputo, configure un túnel entre el nodo de cómputo y el nodo de inicio de sesión como:

```

ssh -N -L localhost:8889:localhost:8889 yourUser@nodeXXXX-X.hpc.iter.es

```

y entre su máquina local y el nodo de login:

```

ssh -N -L localhost:8889:localhost:8889 yourUser@loginX.hpc.iter.es

```

Para acceder a Jupyter Notebook, navegue <http://localhost:8889/>

Notas Importantes

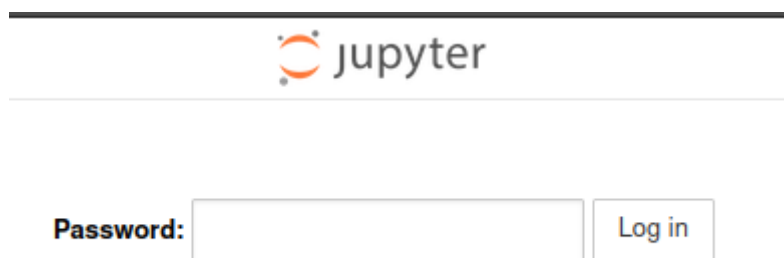
■ Establecer una contraseña segura en Jupyter Notebook

Antes de conectarse a un servidor remoto con jupyter notebook, asegúrese de haber configurado jupyter con información de contraseña. Puede hacerlo editando **jupyter-notebook_config.json** que generalmente se encuentra en `~/.jupyter` o escribiendo:

```
jupyter notebook password
```

```
Enter password:  
Verify password:  
[NotebookPasswordApp] Wrote hashed password to /home/yourUser/.jupyter/  
jupyter_notebook_config.json
```

Podrá acceder ahora con contraseña en lugar de autenticación de token en <http://localhost:8889/login>



The screenshot shows the Jupyter login interface. At the top, there is a horizontal line followed by the Jupyter logo (an orange circle with a white 'j') and the word 'jupyter' in a sans-serif font. Below this, there is a horizontal line. Underneath the line, the word 'Password:' is followed by a text input field. To the right of the input field is a button labeled 'Log in'.

■ Más consejos de seguridad aquí

Lea sobre la seguridad de Jupyter Notebook [aquí](#) y [aquí](#)

■ Cómo cambiar el directorio de inicio de Jupyter Notebook

De forma predeterminada, Jupyter usa su hogar como directorio de inicio predeterminado, pero es posible cambiarlo mediante dos métodos:

- Usando el argumento **--notebook-dir**:

```
jupyter-notebook --no-browser --port=8089 --ip=127.0.0.1 --notebook-dir=/home/my-user/data/my-dir
```

- Cambiar el directorio predeterminado generando un archivo de configuración
- Escriba el siguiente comando para crear una carpeta de configuración.

```
jupyter notebook --generate-config
```

- Abra el archivo `~/jupyter/jupyter_notebook_config.py`.*
- Busque el comentario, *The directory to use for notebooks and kernels*
- Descomente y reemplace la siguiente propiedad en el archivo con su directorio preferido.

```
## The directory to use for notebooks and kernels.  
# Default: "  
c.NotebookApp.notebook_dir = '/home/my-user/data/my_notebooks'
```

- Vuelva a ejecutar Jupyter Notebook.

■ Asegúrese de cerrar su servidor de Jupyter cuando haya terminado

Para hacer esto, si está utilizando el modo *salloc*, puede volver a iniciar sesión en la sesión de *screen* que inició anteriormente donde se ejecuta el cuaderno jupyter y use *ctrl-C* para cerrar el cuaderno jupyter y *salir* para cerrar la sesión con el nodo. Si está utilizando el *sbatch*, puede usar el comando *scancel* .

Conceptos básicos de Jupyter Notebook

En el siguiente enlace puede encontrar una descripción de Jupyter Notebooks:

- [Notebook basics](#)

El dashboard Notebook

To import a notebook, drag the file onto the listing below or [click here](#).

New 

<input type="checkbox"/>	 / examples
<input type="checkbox"/>	 ..
<input type="checkbox"/>	 Builtin Extensions
<input type="checkbox"/>	 Customization
<input type="checkbox"/>	 Embedding
<input type="checkbox"/>	 IPython Kernel
<input type="checkbox"/>	 Interactive Widgets
<input type="checkbox"/>	 Notebook
<input type="checkbox"/>	 Parallel Computing
<input type="checkbox"/>	 images
<input type="checkbox"/>	 utils
<input type="checkbox"/>	 Index.ipynb

Ejemplo básico

```
# This is my first notebook
## Second title
### Thirds title
```

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

```
'''
print("Hellow world")
print(4)
'''
```

```
In [ ]: # This is a comment #

#This is a code cell.

# Execute with ctrl+enter. Try.

print("Hellow world")
print(4)
```

Ejecutar de un Notebook



Not Trusted

Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help

Run Code

This is my first notebook

Second title

Thirds title

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

ejemp

```
In [1]: # This is a comment #  
#This is a code cell.  
# Execute with ctrl+enter. Try.  
print("Hellow world")  
print(4)
```

```
Hellow world  
4
```

Currently running Jupyter processes



Terminals ▾

There are no terminals running.

Notebooks ▾

 data/jupyter_examples/my_first_notebook.ipynb

Python 3 (ipykernel)

Shutdown

hace unos segundos