



Manual de acceso y envío de jobs en GRID-UNAM

(Jupyter + S3/MinIO + HTCondor)

Alicia de la Mora Cebada
Eduardo Ortega Alarcón
DGTIC(UNAM)

Alcance. Acceder al Hub (Jupyter), autenticar la *terminal* del Notebook, habilitar acceso a S3/MinIO y enviar un job de prueba con HTCondor, dejando elementos en S3.

Índice

1. Prerrequisitos	1
2. Ingreso al Hub (Jupyter)	1
3. Asociar la terminal a tu autenticación	4
4. Verificar acceso a S3 desde la terminal	7
5. Enviar un job de prueba con HTCondor	7
5.1. Ejecutable simple: <code>hello.sh</code>	7
5.2. Archivo de envío: <code>hello.sub</code>	8
5.3. Envío del job	8
6. ¿Dónde quedan las salidas?	9
7. Verificación rápida	9
8. Solución de problemas	11
9. Buenas prácticas	11
A. Plantillas listas para copiar	11

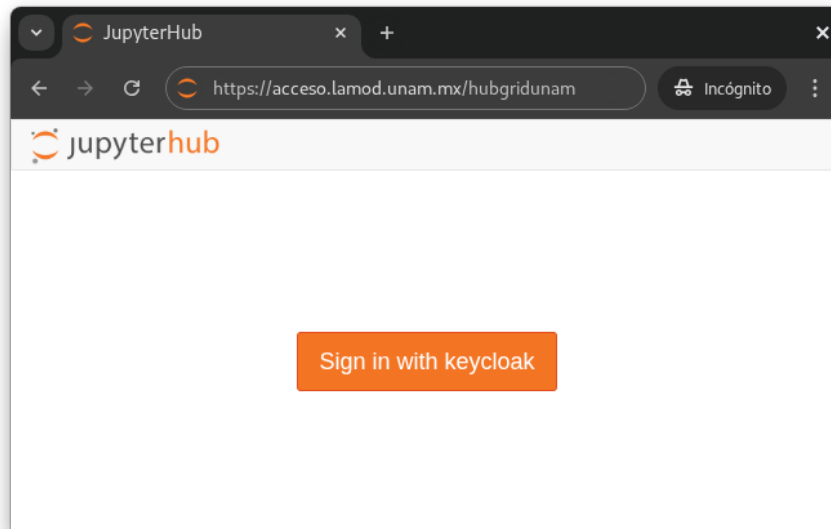
1. Prerrequisitos

- Usuario y contraseña Grid UNAM vigentes.
- Conectividad a internet.

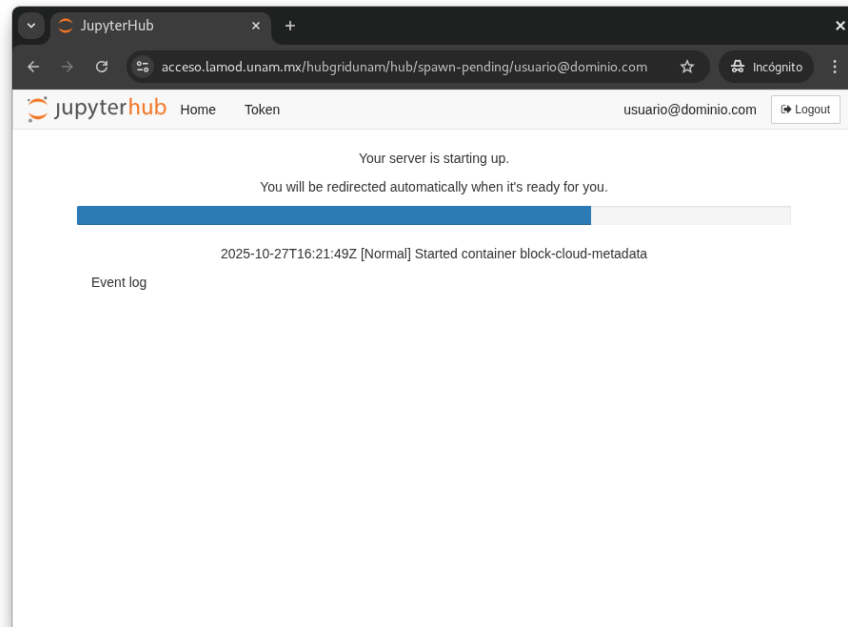
Si tu sesión estuvo inactiva, volver a autenticar *antes* de usar S3 o de enviar trabajos.

2. Ingreso al Hub (Jupyter)

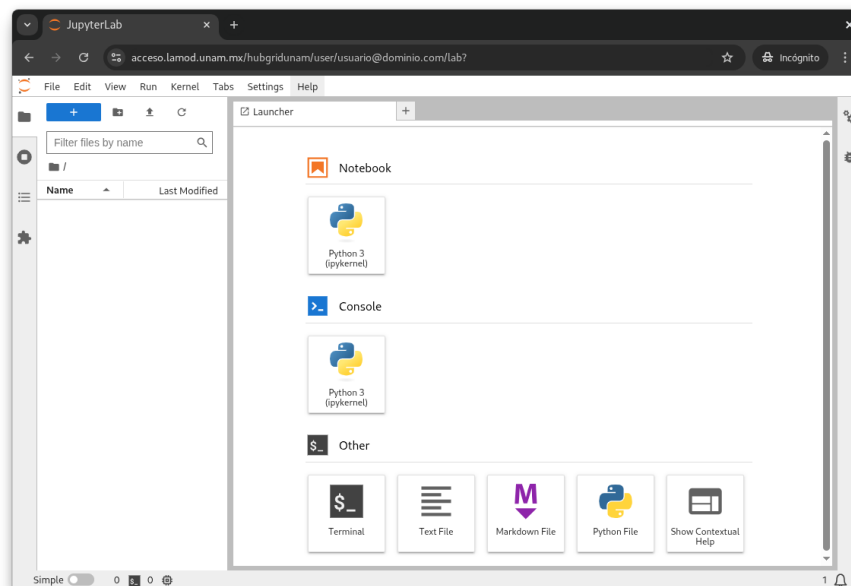
1. Abre: <https://acceso.lamod.unam.mx/hubgridunam/>

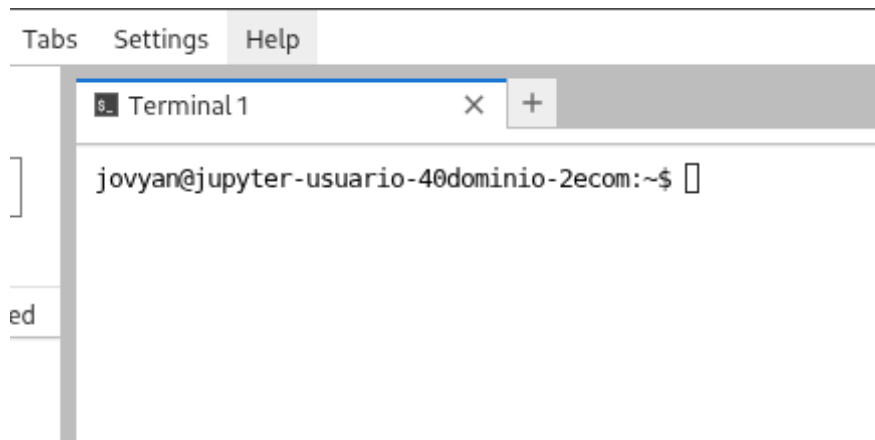


2. Presiona el botón de autenticar denominado como **Sign in with Keycloak**. Esta autenticación se asociará a la terminal en el paso 3.
3. Se ingresan los datos de la cuenta para iniciar sesión *usuario* y *contraseña*.



De esta forma se tendrá acceso al HUB de Grid UNAM (Jupyter) con terminal y shell disponible.



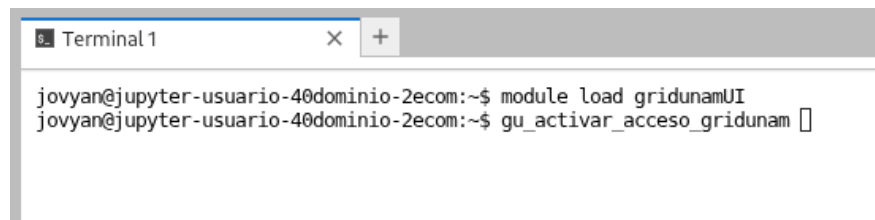


3. Asociar la terminal a tu autenticación

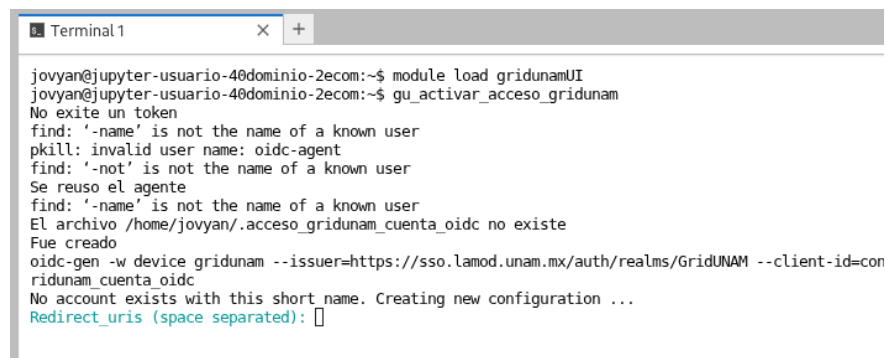
En una **Terminal** del Notebook ejecuta:

Listing 1: Cargar modulo y asociar la terminal

```
module load gridunamUI
gu_activar_acceso_gridunam
```



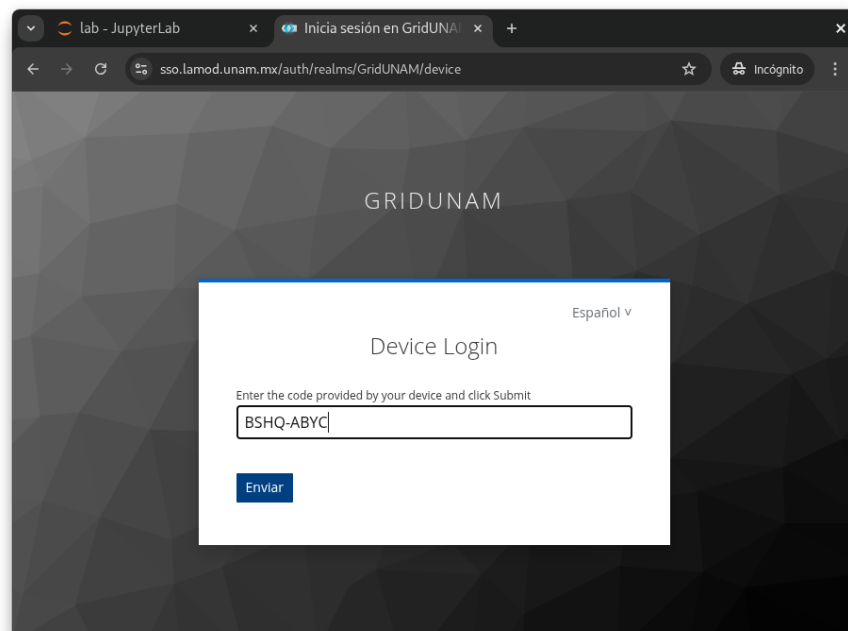
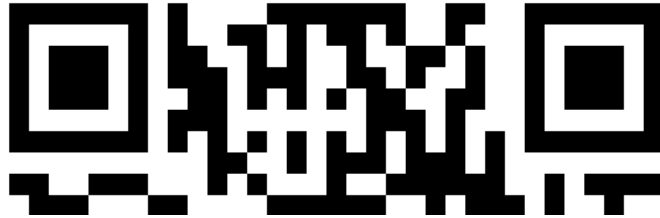
1. Al aparecer el flujo de *device code*, presiona *Enter* donde aparece en color azul **Redirect_uris** (*space separated*) **:**.



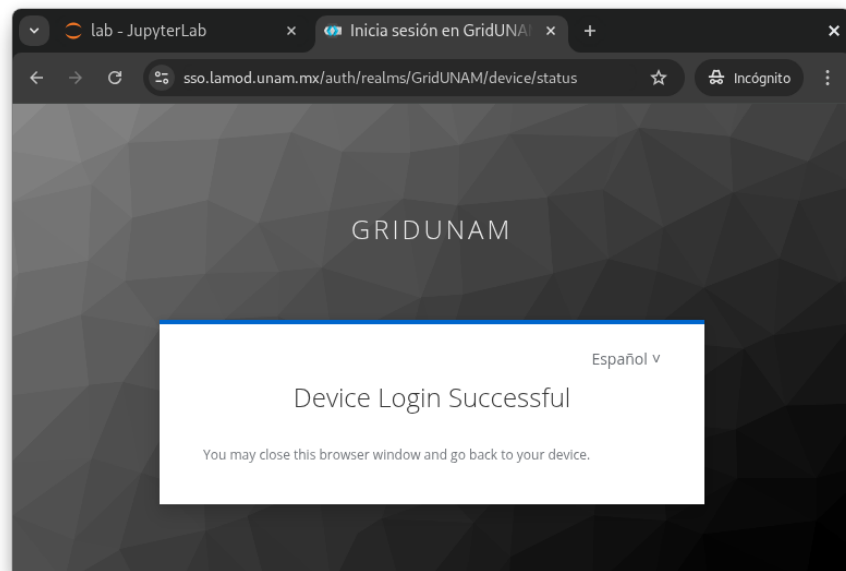
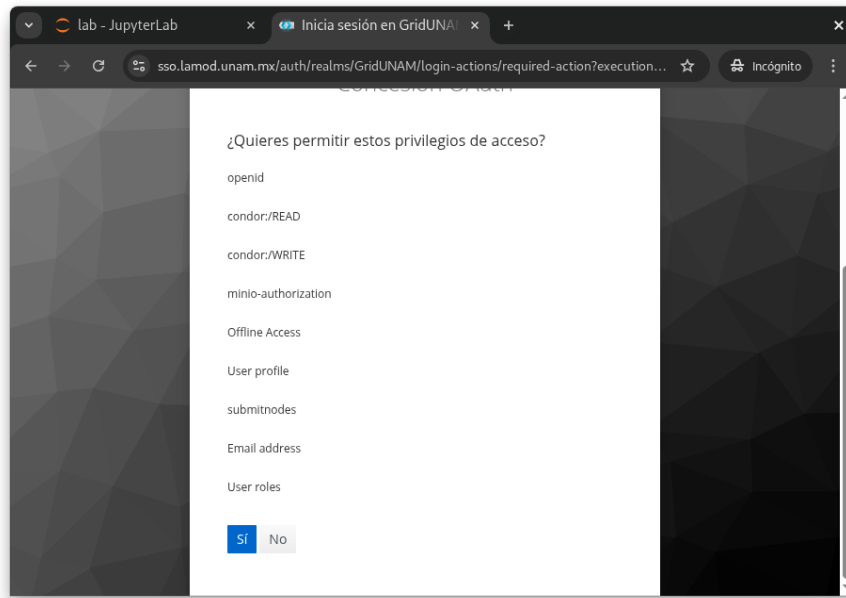
2. Abre en el navegador la URL mostrada y pega el *código* (o escanea el QR).

Using a browser on any device, visit:
<https://sso.lamod.unam.mx/auth/realms/GridUNAM/device>

And enter the code: BSHQ-ABYC
Alternatively you can use the following QR code to visit the above listed URL.



3. Acepta los *scopes* solicitados.



4. Regresa a la terminal y verifica el mensaje de *autenticación correcta*.

```
Everything setup correctly!
Se genero uno nuevo
jovyan@jupyter-usuario-40dominio-2ecom:~$
```

5. Ejecute el comando:

Listing 2: Autenticarse en el almacenamiento

```
. /home/shared/devel/gridauth.sh
```

```
Terminal 1
jovyan@jupyter-usuario-40dominio-2ecom:~$ . /home/shared/devel/gridauth.sh
jovyan@jupyter-usuario-40dominio-2ecom:~$
```

Si el proceso termina sin errores, tu *terminal* queda autenticada y podrás someter trabajos a la Grid, así como acceder al almacenamiento S3/MinIO.

Si no fuera este el caso, entonces habría que hacer lo siguiente y volver a empezar toda esta sección.

Listing 3: Limpieza de sesión

```
cd
module purge
module load gridunamUI
eliminar_mis_agentes
rm -rf .acceso_gridunam_cuenta_oidc .oidc-agent/ /tmp/*
module purge
module load gridunamUI
gu_activar_acceso_gridunam
```

4. Verificar acceso a S3 desde la terminal

El script configura alias/variables para el cliente de S3 (ej. `mcli`). Prueba listando buckets:

Listing 4: Prueba rápida con `mcli`

```
mcli ls dgtic/
```

Los alias pueden variar (`dgtic/`, `lamod/`, `atenea/`). Si no listan, repite la autenticación del paso 3.

```
Terminal 1
jovyan@jupyter-usuario-40dominio-2ecom:~$ . /home/shared/devel/gridauth.sh
jovyan@jupyter-usuario-40dominio-2ecom:~$ mcli ls dgtic/
[2025-10-27 16:11:37 UTC] 0B mi-bucket/
jovyan@jupyter-usuario-40dominio-2ecom:~$
```

5. Enviar un job de prueba con HTCondor

5.1. Ejecutable simple: `hello.sh`

Listing 5: Script de prueba `hello.sh`

```
cat > hello.sh << 'EOF'
#!/bin/bash
echo "Hola, GRID-UNAM"
date '+Listo: %F %T'
```


5.2. Archivo de envío: hello.sub

Incluye la línea de `environment` con variables para S3 y activa el modo remoto:

Listing 6: Submit de HTCondor hello.sub

```
universe = vanilla

environment = "MC_HOST_dgtic=$ENV(MC_HOST_dgtic) MC_HOST_lamod=$ENV(MC_HOST_lamod)
             MC_HOST_atenea=$ENV(MC_HOST_atenea)"

executable = hello.sh
output = salida.%(Cluster).out
error = error.%(Cluster).err
log = log.%(Cluster).log

+remote_jobuniverse = 5
+remote_requirements = True
+remote_ShouldTransferFiles = "YES"
+remote_WhenToTransferOutput = "ON_EXIT"

+xcount = 1
queue
```

La directiva `+xcount` es la responsable de cambiar el número de procesadores.

5.3. Envío del job

Usa el identificador institucional:

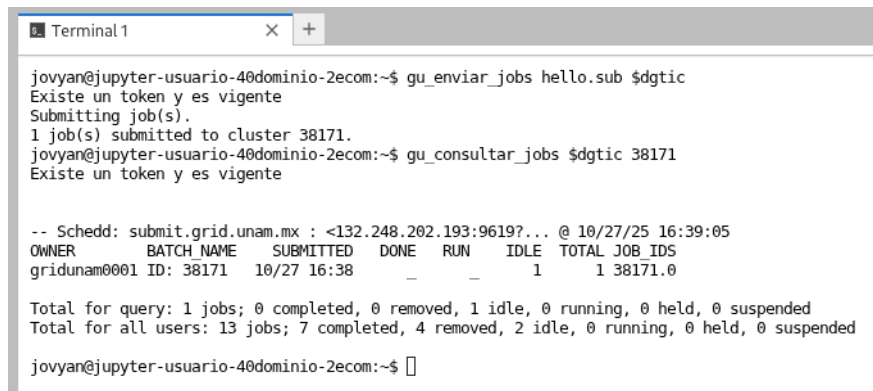
Listing 7: Enviar el trabajo con el identificador

```
gu_enviar_jobs hello.sub $dgtic
# Sustituye $dgtic por el cluster remoto que corresponda
```

Para la consulta del estado del trabajo

Listing 8: Enviar el trabajo con el identificador

```
gu_consultar_jobs $dgtic No.id
```



```
Terminal 1
jovyan@jupyter-usuario-40dominio-2ecom:~$ gu_enviar_jobs hello.sub $dgtic
Existe un token y es vigente
Submitting job(s).
1 job(s) submitted to cluster 38171.
jovyan@jupyter-usuario-40dominio-2ecom:~$ gu_consultar_jobs $dgtic 38171
Existe un token y es vigente

-- Schedd: submit.grid.unam.mx : <132.248.202.193:9619?... @ 10/27/25 16:39:05
OWNER      BATCH NAME    SUBMITTED   DONE    RUN    IDLE  TOTAL JOB_IDS
gridunam0001 ID: 38171   10/27 16:38  _      _      1      1 38171.0

Total for query: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
Total for all users: 13 jobs; 7 completed, 4 removed, 2 idle, 0 running, 0 held, 0 suspended

jovyan@jupyter-usuario-40dominio-2ecom:~$
```

6. ¿Dónde quedan las salidas?

Para descargar los archivos resultantes:

Listing 9: Obtener salida y error del trabajo

```
gu_descargar_archivos $dgtic No.id
```

```
Terminal1
jovyan@jupyter-usuario-40dominio-2ecom:~$ gu_descargar_archivos $dgtic 38171
Existe un token y es vigente
Fetching data files...
jovyan@jupyter-usuario-40dominio-2ecom:~$ ls -l *.38171.*
-rw-r--r-- 1 jovyan users  0 Oct 27 16:41 error.38171.err
-rw-r--r-- 1 jovyan users 846 Oct 27 16:41 log.38171.log
-rw-r--r-- 1 jovyan users 16 Oct 27 16:41 salida.38171.out
jovyan@jupyter-usuario-40dominio-2ecom:~$
```

- Abre `salida.*.out` y confirma que aparece el mensaje del script.
- Verifica que exista `resultado.txt` y, en su caso, que se haya copiado a S3.

```
Terminal1
jovyan@jupyter-usuario-40dominio-2ecom:~$ gu_enviar_jobs hello.sub $dgtic
Existe un token y es vigente
Submitting job(s).
1 job(s) submitted to cluster 38169.
jovyan@jupyter-usuario-40dominio-2ecom:~$ gu_consultar_jobs $dgtic 38169
Existe un token y es vigente

-- Schedd: submit.grid.unam.mx : <132.248.202.193:9619?... @ 10/27/25 16:38:02
OWNER      BATCH NAME    SUBMITTED   DONE    RUN    IDLE  TOTAL JOB_IDS
gridunaml100 ID: 38169   10/27 16:37  -      -      1      1 38169.0

Total for query: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
Total for all users: 11 jobs; 5 completed, 4 removed, 2 idle, 0 running, 0 held, 0 suspended
jovyan@jupyter-usuario-40dominio-2ecom:~$
```

7. Verificación rápida

- La salida estándar (`salida.%(Cluster).out`) y error (`error.%(Cluster).err`) se generan del lado del cluster remoto. Donde `%(Cluster)` representa el número de identificador del trabajo (No. id)
- Para mover elementos a S3 desde el nodo de ejecución:

Listing 10: Copiar elementos a S3/MinIO

```
mcli cp resultado.txt dgtic/mi-bucket/mis-ejecutables/resultado.%(date +%s).txt
# Ajusta dgtic/mi-bucket/... al alias/bucket/ruta válidos en tu entorno
```

Ejemplos: enviar salidas a S3 directamente

A veces conviene mandar la *salida estándar* (`stdout`) directo a un objeto en S3 sin crear archivos temporales. Puedes usar `mcli`:

Listing 11: Piping de stdout hacia S3 con mcli

```
#!/bin/bash

echo "Esta es la salida que se genera a salida estandar... Hola Grid"

# Enviar stdout directamente a S3 (mcli)
echo "Esta es la salida que mando directo al S3" | mcli pipe dgtic/mi-bucket/
    mensajeparaGrid.txt
```

Ejemplo: guardar en archivo local y luego copiar a S3

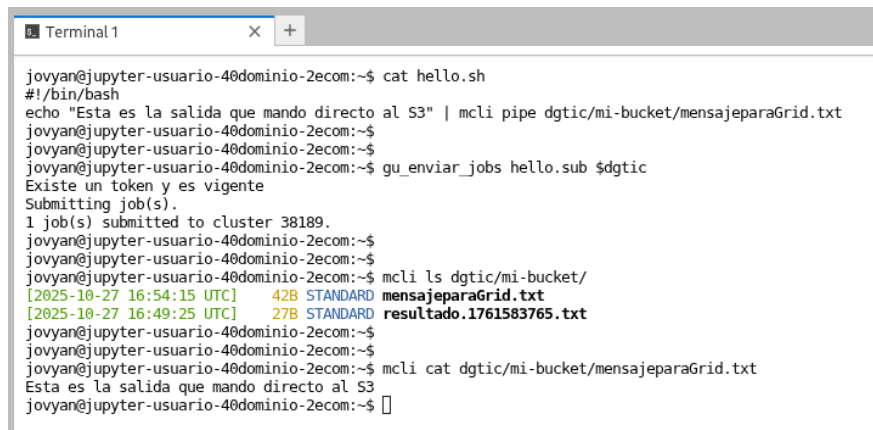
Si prefieres persistir primero localmente y después copiar al bucket/prefijo:

Listing 12: Copiar archivo local a S3

```
# Guardar un mensaje en un archivo local
echo "Este mensaje lo guardo en un archivo que despues copio al S3" > archivo.txt

# Copiar a S3 (mcli)
mcli cp ./archivo.txt dgtic/mi-bucket/
```

Notas. `dgtic/mi-bucket/` es un ejemplo de alias/bucket/prefijo: ajusta el alias (p. ej. `dgtic`, `lamod`) y la ruta a tu caso. El comando `pipe` crea un objeto con el contenido que recibe por `stdin`.



```
Terminal 1
jovyan@jupyter-usuario-40dominio-2ecom:~$ cat hello.sh
#!/bin/bash
echo "Esta es la salida que mando directo al S3" | mcli pipe dgtic/mi-bucket/mensajeparaGrid.txt
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$ gu_enviar_jobs hello.sub $dgtic
Existe un token y es vigente
Submitting job(s).
1 job(s) submitted to cluster 38189.
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$ mcli ls dgtic/mi-bucket/
[2025-10-27 16:54:15 UTC] 42B STANDARD mensajeparaGrid.txt
[2025-10-27 16:49:25 UTC] 27B STANDARD resultado.1761583765.txt
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$ mcli cat dgtic/mi-bucket/mensajeparaGrid.txt
Esta es la salida que mando directo al S3
jovyan@jupyter-usuario-40dominio-2ecom:~$
```

De forma resumida se tiene:

```

Terminal 1
jovyan@jupyter-usuario-40dominio-2ecom:~$ cat hello.sh
#!/bin/bash
echo "Hola, GRID-UNAM"
date '+Listo: %F %T' > resultado.txt
mcli cp resultado.txt dgtic/mi-bucket/resultado.%(date +%s).txt
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$ gu_enviar_jobs hello.sub $dgtic
Existe un token y es vigente
Submitting job(s).
1 job(s) submitted to cluster 38183.
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$ gu_descargar_archivos $dgtic 38183
Existe un token y es vigente
Fetching data files...
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$
jovyan@jupyter-usuario-40dominio-2ecom:~$ ls -l *.38183.*
-rw-r--r-- 1 jovyan users  0 Oct 27 16:50 error.38183.err
-rw-r--r-- 1 jovyan users 175 Oct 27 16:50 log.38183.log
-rw-r--r-- 1 jovyan users 506 Oct 27 16:50 salida.38183.out
jovyan@jupyter-usuario-40dominio-2ecom:~$ cat salida.38183.out
Hola, GRID-UNAM
`/scratch/gridunam0001/37188/resultado.txt` -> `dgtic/mi-bucket/resultado.1761583765.txt`



| Total | Transferred | Duration | Speed      |
|-------|-------------|----------|------------|
| 27 B  | 27 B        | 00m00s   | 1.26 KiB/s |



jovyan@jupyter-usuario-40dominio-2ecom:~$ mcli ls dgtic/mi-bucket/
[2025-10-27 16:49:25 UTC] 27B STANDARD resultado.1761583765.txt
jovyan@jupyter-usuario-40dominio-2ecom:~$ mcli cat dgtic/mi-bucket/resultado.1761583765.txt
Listo: 2025-10-27 10:49:25
jovyan@jupyter-usuario-40dominio-2ecom:~$

```

8. Solución de problemas

Token caducado Re-ejecuta `. /home/shared/devel/gridauth.sh` y repite la *autenticación*.
Sin S3 en jobs Asegúrate de haber autenticado *después* de iniciar sesión en el Hub y de incluir `MC_HOST_*` en `environment` en el archivo `.sub`.

9. Buenas prácticas

- Mantén un bucket/carpeta por proyecto con subcarpetas `inputs/`, `outputs/`, `logs/`.
- Versiona tus archivos `.sub`. Documenta parámetros, datasets.
- Reproduce los experimentos con scripts de análisis y *notebooks* adjuntos.

A. Plantillas listas para copiar

hello.sh

```

#!/bin/bash
echo "Hola, GRID-UNAM"
date '+Listo: %F %T' > resultado.txt

```

hello.sub

```

universe = vanilla

```

```
environment = "MC_HOST_dgtic=$ENV(MC_HOST_dgtic) MC_HOST_lamod=$ENV(MC_HOST_lamod)
    MC_HOST_atenea=$ENV(MC_HOST_atenea)"

executable = hello.sh
output = salida.$(Cluster).out
error = error.$(Cluster).err
log = log.$(Cluster).log

+remote_jobuniverse = 5
+remote_requirements = True
+remote_ShouldTransferFiles = "YES"
+remote_WhenToTransferOutput = "ON_EXIT"

+xcount = 1
queue
```