

Texto en línea

**SLURM**

A.D.G.

Ejercicio 01

Puedes utilizar las MV de los ejercicios anteriores para hacer la siguiente configuración:

compute-0-0:

Tarjeta red modo "Puente" : 10.4.X.Y/8 (DHCP si usas portátil)

Tarjeta de red modo "Red interna" : 172.16.0.10/16

Servidor NIS con usuarios (tunombre1, tunombre2 y grupo tupaellido)

Utiliza NFS para exportar el home de los usuarios

Los usuarios y el root tienen pueden acceder a los nodos por ssh sin utilizar la contraseña

compute-0-1

Tarjeta de red modo "Red interna" : 172.16.0.11/16 (tiene internet a través de compute-0-0)

Cliente NIS

Monta el home de los usuarios con autofs

compute-0-2 (clona de forma enlazada el compute-0-1, cambia nombre /etc/hostname y /etc/host)

Tarjeta de red modo "Red interna" : 172.16.0.12/16 (tiene internet a través de compute-0-0)

Cliente NIS

Monta el home de los usuarios con autofs, recuerda cambiar compute-0-0:/etc/exports

Modificaciones necesarias ahora:

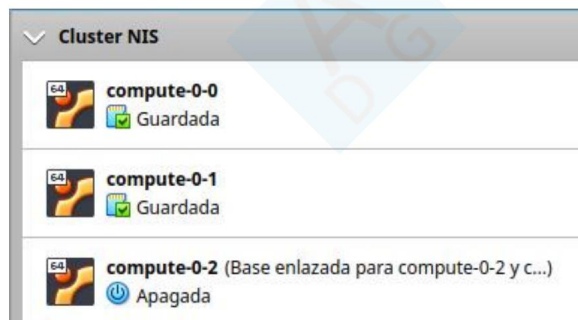
0-0 /etc/exports = añadir las direcciones para 172.16.0.12 de los 4 usuarios también.

0-2 IP en Netplan = modificar a 172.16.0.12/16

0-2 sudo nmtui = cambiar nombre equipo a compute-0-2

0-2 sudo nano /etc/hosts = cambiar nombre de host a compute-0-2

Agrupar los tres clones enlazados en "Cluster NIS"

**SERVIDOR 0-0:**

Instala el gestor de tareas Slurm como se ha visto en la teoría para que los nodos compute-0-1 y compute-0-2 estén en la mismo sistema de colas debug:

```
andreidaniel@compute-0-0:~$ sudo apt-get install slurm-wlm
```

```
andreidaniel@compute-0-0:~$ srange -n | unimunge
STATUS: Success (0)
ENCODING_HOST: compute-0-0 (127.0.1.1)
```

```
andreidaniel@compute-0-0:~$ dmsg -L slurmctld
```

```
andreidaniel@compute-0-0:~$ cd /usr/share/doc/slurmctld/
```

```
andreidaniel@compute-0-0:/usr/share/doc/slurmctld$ python3 -m http.server
```

```
compute-0-0 [Corriendo] - O
andreidaniel@compute-0-0:~$
1: lo: <LOOPBACK> UP,LO
link/loopback 00:00
inet 127.0.0.1/8 sc
valid_lft forever
inet6 ::1/128 scope
valid_lft forever
2: enpos3: <BROADCAST,M
link/ether 08:00:27
inet 192.168.3.105/
valid_lft 86199s
inet6 fe80::800:27
valid_lft forever
3: enpos8: <BROADCAST,M
link/ether 08:00:27
inet 172.16.0.10/16
```

← → ↻ No es seguro | 192.168.3.105:8000

Directory listing for /

IPServidor(Bridge):8000 192.168.3.105:8000. Se accede a:

- [changelog.Debian.gz](#)
- [copyright](#)
- [examples](#)
- [NEWS.Debian.gz](#)
- [README.Debian](#)
- [slurm-wlm-configurator-easy.html](#)
- [slurm-wlm-configurator.html](#)

Se modifican estos apartados:

SlurmctldHost = compute-0-0

NodeName = compute-0-[1-2]

ProctrackType = linuxproc

->Submit

Se accede con MobaXterm y se copia el resultado generado anteriormente en /etc/slurm/slurm.conf:

```
andreidaniel@compute-0-0:~$ cd /etc/slurm/
andreidaniel@compute-0-0:/etc/slurm$ sudo su
root@compute-0-0:/etc/slurm# cat > slurm.conf
# slurm.conf file generated by configurator.html.
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.
#
ClusterName=cluster
SlurmctldHost=compute-0-0
#SlurmctldHost=
#
#DisableRootJobs=NO
#EnforcePartitions=NO
#EpiLog=
#EpiLogSlurmctld=
#FirstJobId=1
#MaxJobId=67843328
#GresTypes=
#GroupUpdateForce=0
#GroupUpdateTime=600
#JobFileAppend=0
#JobRequeue=1
#JobSubmitPlugins=lua
#KillOnBadExit=0
#LaunchType=launch/slurm
```

Se descomenta o copia estas líneas dentro (que no se repitan, y menos Proctrack):

```
SlurmUser=root
SlurmdUser=root
AuthType=auth/munge
CryptoType=crypto/munge
ProctrackType=proctrack/linuxproc
```

Añadir en /etc/hosts a los dos clientes también:

```
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
127.0.0.1 compute-0-0
#añadido & modificado:
172.16.0.10 servidor.AND.nis
172.16.0.11 compute-0-1
172.16.0.12 compute-0-2
```

```
root@compute-0-0:/etc/slurm# systemctl start slurmctld
```

```
root@compute-0-0:/etc/slurm# systemctl restart slurmctld
```

```
root@compute-0-0:/etc/slurm# sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
debug* up infinite 2 unk* compute-0-[1-2]
```

CLIENTE 0-1

```
andreidaniel@compute-0-1:/$ sudo apt-get install slurmd
```

Comprobar que los nodos son accesibles por el root desde el servidor sin el uso de contraseña:

```
andreidaniel@compute-0-0:/etc/slurm$ sudo su
[sudo] password for andreidaniel:
root@compute-0-0:/etc/slurm# ssh compute-0-1 hostname
The authenticity of host 'compute-0-1 (172.16.0.11)' can't be established.
ED25519 key fingerprint is SHA256:2Pn0+YYQxqHeX+RLZhQQUicUNAdr0di9syFupLHbu0.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'compute-0-1' (ED25519) to the list of known hosts.
compute-0-1
```

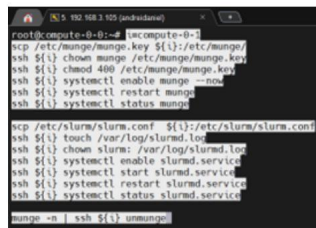
```
andreidaniel@compute-0-1:/$ munge -n | unmunge
STATUS:      Success (0)
ENCODE_HOST: compute-0-1 (127.0.1.1)
```

En el Servidor 0-0 se copia tal cual (no necesario script o archivo) y dará el siguiente resultado:

```
l@compute-0-1
scp /etc/munge/munge.key ${!}/etc/munge/
ssh ${!} chown munge /etc/munge/munge.key
ssh ${!} chmod 400 /etc/munge/munge.key
ssh ${!} systemctl enable munge --now
ssh ${!} systemctl restart munge
ssh ${!} systemctl status munge

scp /etc/slurm/slurm.conf ${!}/etc/slurm/slurm.conf
ssh ${!} touch /var/log/slurmd.log
ssh ${!} chown slurm: /var/log/slurmd.log
ssh ${!} systemctl enable slurmd.service
ssh ${!} systemctl start slurmd.service
ssh ${!} systemctl restart slurmd.service
ssh ${!} systemctl status slurmd.service

munge -n | ssh ${!} unmunge
```



```
andreidaniel@compute-0-1:~$ cat /etc/munge/munge.key
andreidaniel@compute-0-1:~$ scp /etc/munge/munge.key root@compute-0-0:/etc/munge/
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'chown munge /etc/munge/munge.key'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'chmod 400 /etc/munge/munge.key'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'systemctl enable munge --now'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'systemctl restart munge'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'systemctl status munge'
andreidaniel@compute-0-1:~$ scp /etc/slurm/slurm.conf root@compute-0-0:/etc/slurm/
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'touch /var/log/slurmd.log'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'chown slurm: /var/log/slurmd.log'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'systemctl enable slurmd.service'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'systemctl start slurmd.service'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'systemctl restart slurmd.service'
andreidaniel@compute-0-1:~$ ssh root@compute-0-0 'systemctl status slurmd.service'
andreidaniel@compute-0-1:~$ munge -n | ssh root@compute-0-0 'unmunge'
```

```

6. 192.168.3.105 (andreidaniel)
[sudo] password for andreidaniel:
root@compute-0-0:/home/andreidaniel# i=compute-0-1
scp /etc/munge/munge.key ${i}:/etc/munge/
ssh ${i} chown munge /etc/munge/munge.key
ssh ${i} chmod 400 /etc/munge/munge.key
ssh ${i} systemctl enable munge --now
ssh ${i} systemctl restart munge
ssh ${i} systemctl status munge

scp /etc/slurm/slurm.conf ${i}:/etc/slurm/slurm.conf
ssh ${i} touch /var/log/slurmd.log
ssh ${i} chown slurm: /var/log/slurmd.log
ssh ${i} systemctl enable slurmd.service
ssh ${i} systemctl start slurmd.service
ssh ${i} systemctl restart slurmd.service
ssh ${i} systemctl status slurmd.service

munge -n | ssh ${i} unmunge
munge.key
100% 128 85.3KB/s 00:00
Synchronizing state of munge.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable munge
● munge.service - MUNGE authentication service
   Loaded: loaded (/lib/systemd/system/munge.service; enabled; vendor preset: enable
   Active: active (running) since Wed 2023-01-25 13:00:47 UTC; 314ms ago
     Docs: man:munged(8)
   Process: 7162 ExecStart=/usr/sbin/munged $OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 7164 (munged)
       Tasks: 4 (limit: 2238)
      Memory: 816.0K
         CPU: 13ms
        CGroup: /system.slice/munge.service
                └─7164 /usr/sbin/munged

ene 25 13:00:47 compute-0-1 systemd[1]: Starting MUNGE authentication service...
ene 25 13:00:47 compute-0-1 systemd[1]: Started MUNGE authentication service.
slurm.conf
100% 3006 1.4MB/s 00:00
Synchronizing state of slurmd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable slurmd
● slurmd.service - Slurm node daemon
   Loaded: loaded (/lib/systemd/system/slurmd.service; enabled; vendor preset: enable
   Active: active (running) since Wed 2023-01-25 13:00:50 UTC; 306ms ago
     Docs: man:slurmd(8)
   Main PID: 7617 (slurmd)
       Tasks: 1
      Memory: 1.1M
         CPU: 78ms
        CGroup: /system.slice/slurmd.service
                └─7617 /usr/sbin/slurmd -D -s

ene 25 13:00:50 compute-0-1 systemd[1]: Started Slurm node daemon.
ene 25 13:00:50 compute-0-1 slurmd[7617]: slurmd: slurmd version 21.08.5 started
ene 25 13:00:50 compute-0-1 slurmd[7617]: slurmd: slurmd started on Wed, 25 Jan 2023 1
3:00:50 +0000
ene 25 13:00:50 compute-0-1 slurmd[7617]: slurmd: CPUs=1 Boards=1 Sockets=1 Cores=1 Th
reads=1 Memory=1976 TmpDisk=9979 Uptime=9971 CPUSpecList=(null) FeaturesAvail=(null) F
eaturesActive=(null)
STATUS: Success (0)
ENCODE_HOST: compute-0-1 (127.0.1.1)
ENCODE_TIME: 2023-01-25 13:00:50 +0000 (1674651650)
DECODE_TIME: 2023-01-25 13:00:51 +0000 (1674651651)
TTL: 300
CIPHER: aes128 (4)
MAC: sha256 (5)
ZIP: none (0)
UID: root (0)
GID: root (0)
LENGTH: 0
root@compute-0-0:/home/andreidaniel#

```

```

root@compute-0-0:~# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*    up       infinite    1    unk*  compute-0-2
debug*    up       infinite    1    idle  compute-0-1

```

Por si no se ha iniciado bien (instalar scontrol): scontrol update nodename=compute-0-1 state=idle

```
root@compute-0-1:~# sudo apt-get install slurm-client
```

 (scontrol) Volverá a estar iniciado en "sinfo".

Se hace también para el Cliente 0-2

CLIENTE 0-2

```
andreidaniel@compute-0-2:/$ sudo apt-get install slurmd
```

Comprobar que los nodos son accesibles por el root desde el servidor sin el uso de contraseña:

```

root@compute-0-0:~# ssh 172.16.0.12 hostname
The authenticity of host '172.16.0.12 (172.16.0.12)' can't be established.
ED25519 key fingerprint is SHA256:2Pn0+YYQxkqHeX+RLZhQQUiciNAdR0d19syFupLHbu0.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.0.12' (ED25519) to the list of known hosts.
compute-0-2

```

En el Servidor 0-0 se copia tal cual (no necesario script o archivo) y dará el siguiente resultado:

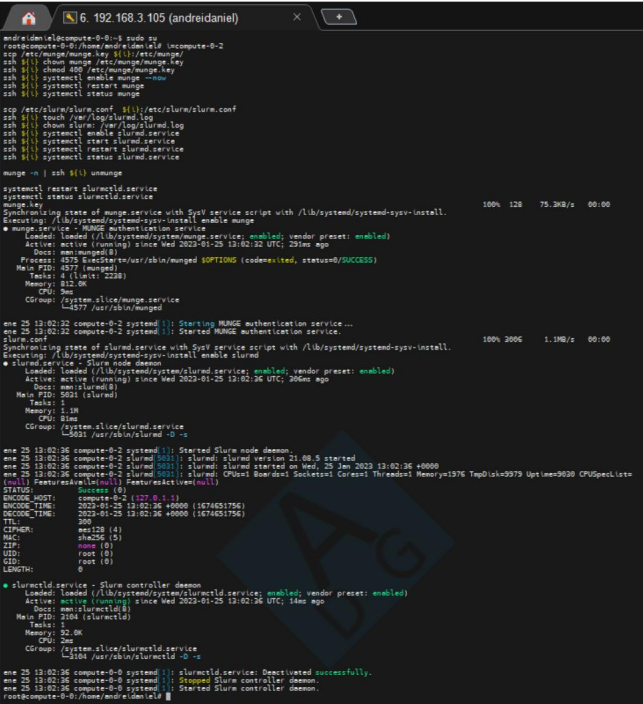
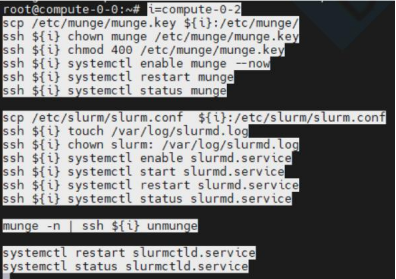

```
i=compute-0-2
scp /etc/munge/munge.key ${i}:/etc/munge/
ssh ${i} chown munge /etc/munge/munge.key
ssh ${i} chmod 400 /etc/munge/munge.key

ssh ${i} systemctl enable munge --now
ssh ${i} systemctl restart munge
ssh ${i} systemctl status munge

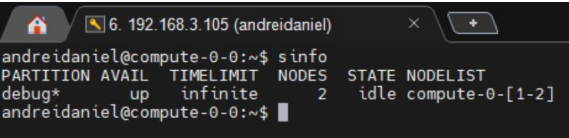
scp /etc/slurm/slurm.conf ${i}:/etc/slurm/slurm.conf
ssh ${i} touch /var/log/slurmd.log
ssh ${i} chown slurm: /var/log/slurmd.log
ssh ${i} systemctl enable slurmd.service
ssh ${i} systemctl start slurmd.service
ssh ${i} systemctl restart slurmd.service
ssh ${i} systemctl status slurmd.service

munge -n | ssh ${i} unmunge

systemctl restart slurmctld.service
systemctl status slurmctld.service
```



quedaría como:



Por si no se ha iniciado bien (instalar scontrol): scontrol update nodename=compute-0-2 state=idle

root@compute-0-2:~# sudo apt-get install slurm-client (scontrol) Volverá a estar iniciado en "sinfo".

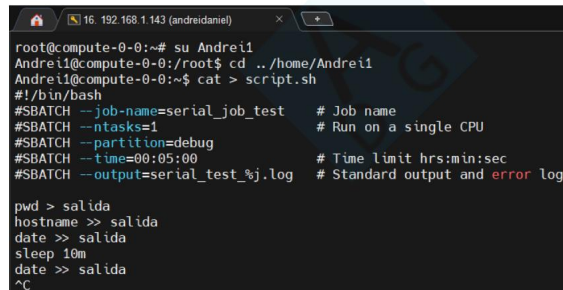
Crea el script.sh en el usuario tunombre1:

Servidor 0-0:

Se copia el script en un archivo script.sh:

```
#!/bin/bash
#SBATCH --job-name=serial_job_test # Job name
#SBATCH --ntasks=1 # Run on a single CPU
#SBATCH --partition=debug
#SBATCH --time=00:05:00 # Time limit hrs:min:sec
#SBATCH --output=serial_test_%j.log # Standard output and error log
```

```
pwd > salida
hostname >> salida
date >> salida
sleep 10m
date >> salida
```



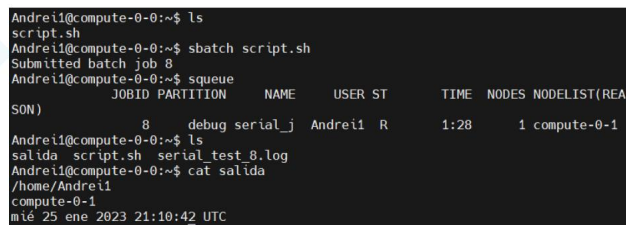
```
root@compute-0-0:~# su Andrei1
Andrei1@compute-0-0:/root$ cd ../home/Andrei1
Andrei1@compute-0-0:~$ cat > script.sh
#!/bin/bash
#SBATCH --job-name=serial_job_test # Job name
#SBATCH --ntasks=1 # Run on a single CPU
#SBATCH --partition=debug
#SBATCH --time=00:05:00 # Time limit hrs:min:sec
#SBATCH --output=serial_test_%j.log # Standard output and error log

pwd > salida
hostname >> salida
date >> salida
sleep 10m
date >> salida
^C
```

Por si los dos nodos no están iniciados (pues no nos funcionaría el ejecutador sbatch siguiente) se ejecuta otra vez el código ese anterior (i=compute-0-1 ... luego con el -0-2...)

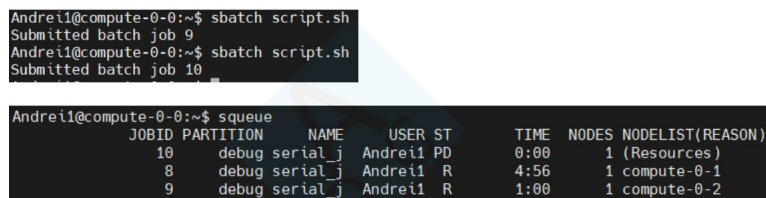
Lanzalo con sbatch y haz un pantallazo de como se ejecuta en el sistema slurm.

Ejemplos de como funciona el sistema slurm:



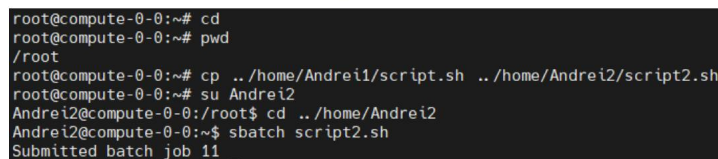
```
Andrei1@compute-0-0:~$ ls
script.sh
Andrei1@compute-0-0:~$ sbatch script.sh
Submitted batch job 8
Andrei1@compute-0-0:~$ squeue
        JOBID PARTITION    NAME    USER  ST       TIME  NODES NODELIST(REAS
SON)
             8      debug serial_j  Andrei1  R         1:28      1 compute-0-1
Andrei1@compute-0-0:~$ ls
salida script.sh serial_test_8.log
Andrei1@compute-0-0:~$ cat salida
/home/Andrei1
compute-0-1
mié 25 ene 2023 21:10:42 UTC
```

Fíjate lo que sucede si lanzamos más trabajos:



```
Andrei1@compute-0-0:~$ sbatch script.sh
Submitted batch job 9
Andrei1@compute-0-0:~$ sbatch script.sh
Submitted batch job 10
Andrei1@compute-0-0:~$ squeue
        JOBID PARTITION    NAME    USER  ST       TIME  NODES NODELIST(REASON)
             9      debug serial_j  Andrei1  R         4:56      1 compute-0-1
            10      debug serial_j  Andrei1  PD         0:00      1 (Resources)
             8      debug serial_j  Andrei1  R         1:00      1 compute-0-2
```

Lo lanzamos con otro usuario:



```
root@compute-0-0:~# cd
root@compute-0-0:~# pwd
/root
root@compute-0-0:~# cp ../home/Andrei1/script.sh ../home/Andrei2/script2.sh
root@compute-0-0:~# su Andrei2
Andrei2@compute-0-0:/root$ cd ../home/Andrei2
Andrei2@compute-0-0:~$ sbatch script2.sh
Submitted batch job 11
```

IMPORTANTE: Como escribí CANCEL 16 y falló el comando por las mayúsculas, o luego puse cancel 16 pero no había job con id 16 y por eso ya si fue seguramente, se borraron los jobs, ya no aparecían en squeue, por tanto volví a iniciar 1 trabajo con Andrei1 y 2 trabajos con Andrei2 ejecutando las veces respectivas el script.sh y luego:

Cancelamos un trabajo:

```
16. 192.168.1.143 (andreidaniel)
Andrei2@compute-0-0:~$ squeue
      JOBID PARTITION  NAME   USER ST      TIME  NODES NODELIST(REASON)
       14      debug  serial_j Andrei2 PD       0:00      1 (Resources)
       12      debug  serial_j Andrei1 R       0:46      1 compute-0-1
       13      debug  serial_j Andrei2 R       0:14      1 compute-0-2
Andrei2@compute-0-0:~$ scancel 13
Andrei2@compute-0-0:~$ squeue
      JOBID PARTITION  NAME   USER ST      TIME  NODES NODELIST(REASON)
       12      debug  serial_j Andrei1 R       1:14      1 compute-0-1
       14      debug  serial_j Andrei2 R       0:03      1 compute-0-2
Andrei2@compute-0-0:~$
```

Comentario

Calificación	2,00 / 2,00
Calificado sobre	jueves, 26 de enero de 2023, 11:03
Calificado por	

◀ Ejercicios NFS

Ir a...

Ejercicios Environment modules ▶