

# Singularity + SLURM



# Problemas con Docker directo



- **Invisibilidad de recursos**  
Los trabajos no aparecen en squeue
- **Monopolización no controlada**  
Un usuario puede tomar toda la GPU/CPU
- **Falta de fairshare**  
No se respetan las políticas de uso justo
- **Conflictos de recursos**  
Múltiples usuarios compitiendo sin coordinación
- **Sin límites de tiempo**  
Trabajos que corren indefinidamente



## ¿Qué es Singularity?

Contenedores diseñados para HPC

- Sin privilegios root: Seguro para clusters multiusuario
- Compatibilidad con Docker: Puede usar imágenes de Docker Hub
- Integración HPC: Diseñado para trabajar con schedulers como SLURM
- Rendimiento: Acceso directo al hardware (GPUs, InfiniBand)
- Reproducibilidad: Entornos consistentes y portables



# Cómo funciona todo junto

## Flujo de trabajo:

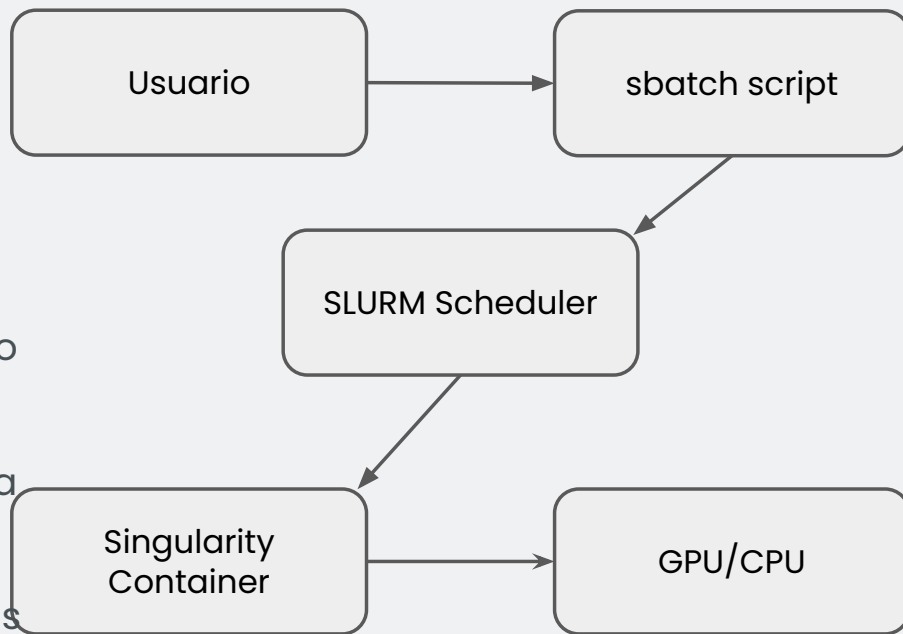
**Usuario prepara:** Script SLURM + imagen Singularity

**SLURM programa:** Asigna recursos según disponibilidad

**Singularity ejecuta:** Contenedor con acceso controlado al hardware

**SLURM monitorea:** Controla límites y registra uso

**Cleanup automático:** Liberación de recursos al terminar



# Creando Imágenes Singularity

```
bash
```

```
# Descargar imagen de TensorFlow
```

```
singularity pull tensorflow.sif docker://tensorflow/tensorflow:latest-gpu
```

```
# Desde imagen personalizada
```

```
singularity pull mi_proyecto.sif docker://usuario/mi_imagen:tag
```

```
bash
```

```
# Construir imagen Docker primero
```

```
docker build -t mi_proyecto:latest .
```

```
# Convertir a Singularity
```

```
singularity build mi_proyecto.sif docker-daemon://mi_proyecto:latest
```



# Ejecutando imagenes con SLURM

bash

```
#!/bin/bash
#SBATCH --job-name=pytorch_training
#SBATCH --cpus-per-task=8
#SBATCH --mem=16gb
#SBATCH --output=/raid/home/$USER/logs/pytorch_%j.out
#SBATCH --error=/raid/home/$USER/logs/pytorch_%j.err
#SBATCH --nodelist=dgxa100ja1
#SBATCH --gres=gpu:1
#SBATCH --partition=dgx_small
#SBATCH --time=02:00:00

pwd; hostname; date

# Ejecutar contenedor
singularity run --nv pytorch.sif --epochs 100 --batch-size 32

date
```



# Uso de binds (volúmenes)

```
bash
```

```
#!/bin/bash
#SBATCH --job-name=data_processing
#SBATCH --cpus-per-task=16
#SBATCH --mem=32gb
#SBATCH --output=/raid/home/$USER/logs/processing_%j.out
#SBATCH --nodelist=dgxa100jal
#SBATCH --gres=gpu:1
#SBATCH --partition=dgx_small
#SBATCH --time=04:00:00
```

```
pwd; hostname; date
```

```
# Bind mounts para datos compartidos
singularity run --nv \
  --bind /raid/datasets:/data \
  --bind /raid/home/$USER/results:/results \
  --bind /raid/home/$USER/code:/code \
  mi_proyecto.sif \
  /code/process_data.py --input /data --output /results
```

```
date
```





**IIEG**

**Instituto de Información  
Estadística y Geográfica  
de Jalisco**