



## Tutorial 02 - Usando Contenedores

NOTA: El supercomputador Patagón funciona mediante contenedores, esto permite que cada usuario se *auto-atienda* en cuanto a los paquetes/librerías que necesite. Patagón por defecto soporta contenedores de <https://hub.docker.com/> sin tener que configurar la autenticación. Por ejemplo:

```
srunc --container-image=ubuntu:24.04 --container-name=ubuntu --pty bash
```

Descargará y usará de inmediato un contenedor de Ubuntu 24.04 para luego ejecutar un shell `bash` interactivo dentro. Sin embargo, para usar contenedores de NVIDIA los cuales vienen con paquetes y configuraciones listas para funcionar con GPUs, es necesario configurar el sistema de autenticación. Este tutorial muestra paso a paso como lograr esto para cuando se descargan contenedores al usar `srunc` o `sbatch`.

### 1) Crear archivo de `.credentials`

Entrar a Patagón y crear un archivo oculto en la ubicación `~/.config/enroot/.credentials` con el siguiente contenido:

```
# NVIDIA GPU Cloud (both endpoints are required)
machine nvcr.io login $oauthtoken password APIKEY
machine authn.nvidia.com login $oauthtoken password APIKEY

# DockerHub
# machine auth.docker.io login <login> password <password>

# Google Container Registry
# machine gcr.io login oauth2accesstoken password $(gcloud auth print-access-token)
# machine gcr.io login _json_key password $(jq -c '.' $GOOGLE_APPLICATION_CREDENTIALS | sed 's/ /\u0020/g')
```

### 2) Autenticación para Nvidia GPU Cloud (NGC)

En el caso de Nvidia GPU Cloud (NGC), primero es necesario tener una cuenta de usuario. Si no la tiene, primero ir al [sitio web de NGC](#) y crear una cuenta. Para más información sobre crear una cuenta en NGC, [seguir este tutorial](#).

En este paso vamos a [generar una API KEY](#) la cual copiaremos en el archivo `.credentials`. Para ello, primero ingresamos a Nvidia GPU Cloud con nuestra cuenta, y en el menú seleccionar **setup -->Get API key --> Generate API Key**. La API Key es una cadena de caracteres, por ejemplo `aCbAsbasoi18IPASNQPqpsqsnqnqSQq71SnQQqsMAqqsqsewQsqSQSdqk23`. Copiar la llave y pegarla en su archivo `.credentials` en lugar de `APIKEY`.

Importante

- Cuide su API Key ya que es personal y sirve para autenticarse con su cuenta de usuario.



- Para comprobar el funcionamiento puede probar descargar el contenedor base de CUDA de NGC (puede demorar algunos segundos ya que se comenzara a descargar el contenedor)

```
→ ~ srun --container-name=cuda-12.9 --container-image='nvcr.io/nvidia/cuda:12.9.1-base-ubuntu24.04' nv
pyxis: importing docker image: nvcr.io/nvidia/cuda:12.9.1-base-ubuntu24.04
pyxis: imported docker image: nvcr.io/nvidia/cuda:12.9.1-base-ubuntu24.04
No devices found.
```

Si ve el mismo output, entonces NGC ha quedado bien configurado. Solo queda borrar el contenedor ya que fue solo para testear:

```
→ ~ srun --pty enroot remove pyxis_cuda-12.9
Do you really want to delete /home/enroot-data/user-10082/pyxis_cuda-12.9? [y/N] y
→ ~
```

### 3) Autenticación en DockerHub

Este paso es opcional y solo es necesario cuando necesita un contenedor de DockerHub que requiere de autenticación. Para configurar esta autenticación, debe rellenar la sección correspondiente a DockerHub con su usuario y contraseña. Otra alternativa es hacerlo de forma similar a NGC, generando un ACCESS TOKEN y copiarlo en la sección Docker Hub del archivo `.credentials`. Mas información en [la documentacion de Docker Hub](#).

### 4) Autenticación con Google Container Registry (GCR)

De forma similar a NGC, generar un ACCESS TOKEN y copiarlo en la sección de Google Container Registry del archivo `.credentials` Mas información en [la documentacion de GCR](#).

Universidad Austral de Chile