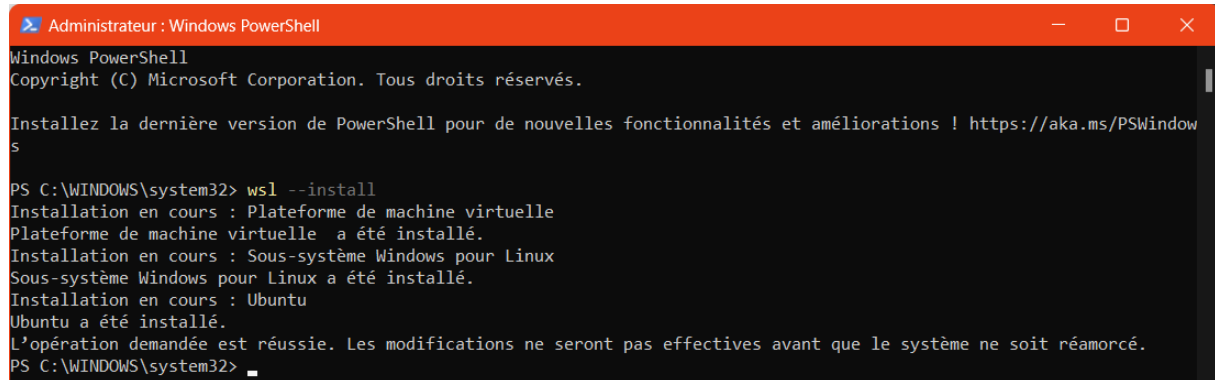


## Containerization Technologies – TD 1

We install wsl (Linux kernel) :

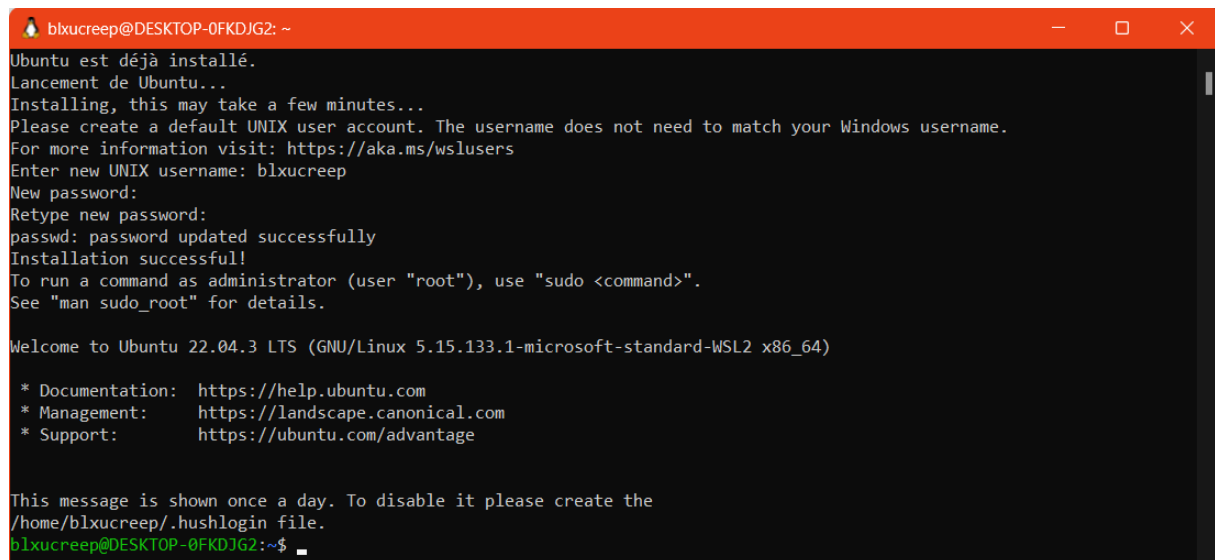


```
Administrateur : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install
Installation en cours : Plateforme de machine virtuelle
Plateforme de machine virtuelle a été installé.
Installation en cours : Sous-système Windows pour Linux
Sous-système Windows pour Linux a été installé.
Installation en cours : Ubuntu
Ubuntu a été installé.
L'opération demandée est réussie. Les modifications ne seront pas effectives avant que le système ne soit réamorcé.
PS C:\WINDOWS\system32>
```

After restarting my PC, it asks for an username and a password:



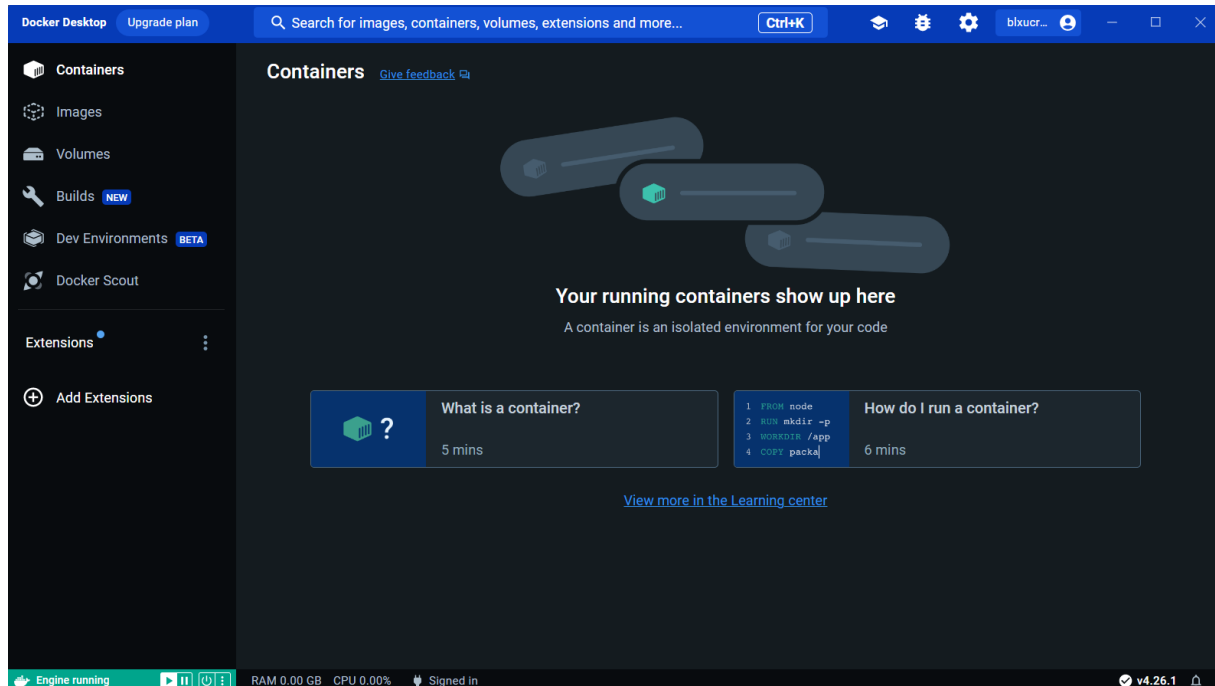
```
blxucreep@DESKTOP-0FKDJG2: ~
Ubuntu est déjà installé.
Lancement de Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: blxucreep
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.133.1-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/blxucreep/.hushlogin file.
blxucreep@DESKTOP-0FKDJG2:~$
```

Docker is installed:



Let's see what commands I can use with Docker:

```
blxucreep@DESKTOP-0FKDJG2: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
blxucreep@DESKTOP-0FKDJG2:~$ docker  
  
Usage: docker [OPTIONS] COMMAND  
  
A self-sufficient runtime for containers  
  
Common Commands:  
run      Create and run a new container from an image  
exec     Execute a command in a running container  
ps       List containers  
build    Build an image from a Dockerfile  
pull     Download an image from a registry  
push     Upload an image to a registry  
images   List images  
login    Log in to a registry  
logout   Log out from a registry  
search   Search Docker Hub for images  
version  Show the Docker version information  
info     Display system-wide information  
  
Management Commands:  
builder  Manage builds  
buildx*  Docker Buildx (Docker Inc., v0.12.0-desktop.2)  
compose* Docker Compose (Docker Inc., v2.23.3-desktop.2)  
container Manage containers  
context  Manage contexts
```

Now we login in Docker:

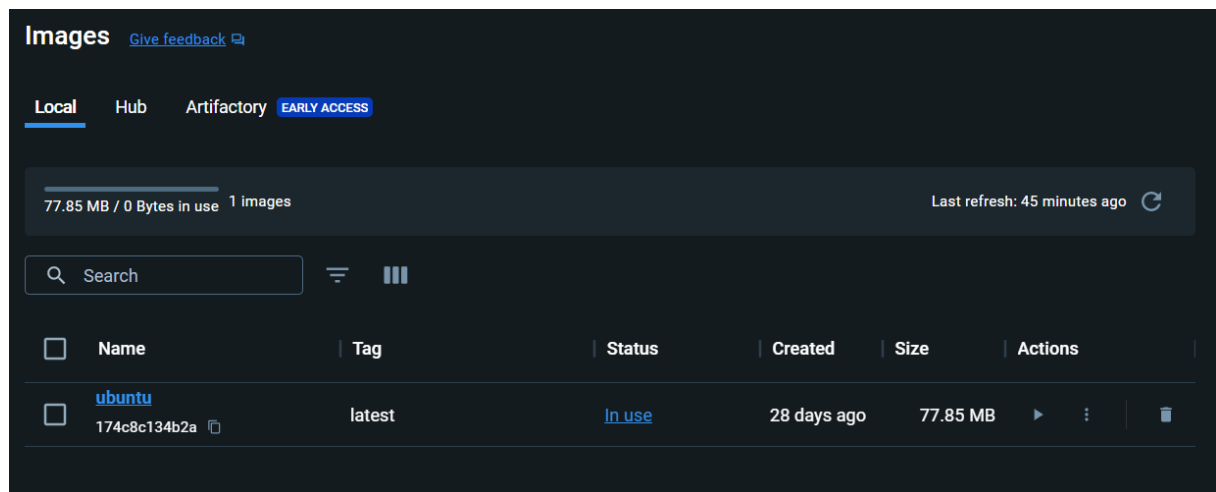
```
blxucreep@DESKTOP-0FKDJG2:~$ docker login  
Authenticating with existing credentials...  
Login Succeeded  
blxucreep@DESKTOP-0FKDJG2:~$
```

Here, like our Windows system, now we need an environment to use our newly created Linux kernel. Let's install the latest Ubuntu distribution from the Docker registry:

```
blxucreep@DESKTOP-0FKDJG2:~$ docker pull ubuntu:latest
latest: Pulling from library/ubuntu
a48641193673: Pull complete
Digest: sha256:6042500cf4b44023ea1894effe7890666b0c5c7871ed83a97c36c76ae560bb9b
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu:latest
blxucreep@DESKTOP-0FKDJG2:~$
```

We can verify this in Docker directly, our image is available to use:



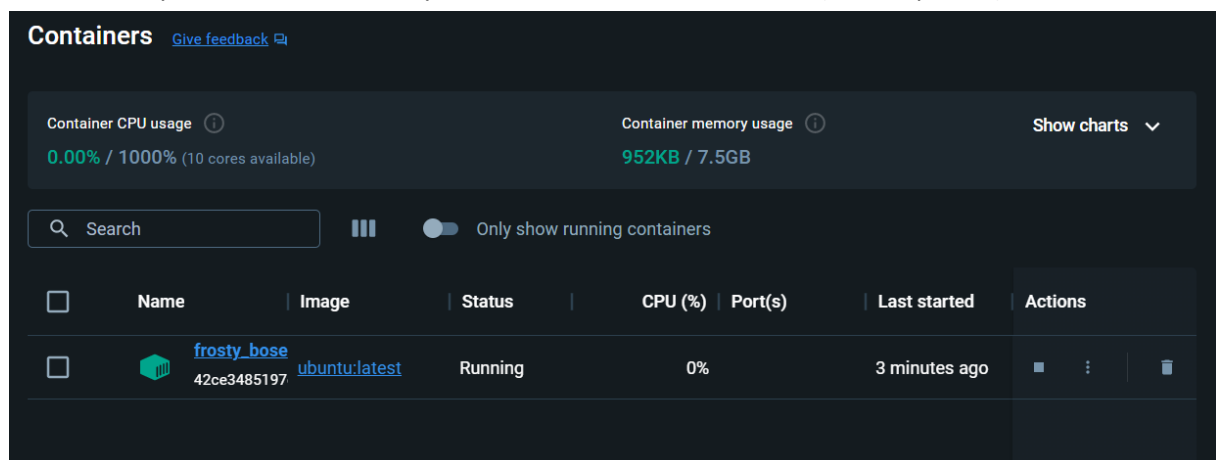
The screenshot shows the 'Images' tab in Docker Desktop. It displays a table with one image, 'ubuntu', which is 'In use'. The image has a size of 77.85 MB and was created 28 days ago. The interface includes a search bar, a filter icon, and a 'Last refresh' timestamp of 45 minutes ago.

Name	Tag	Status	Created	Size	Actions
ubuntu 174c8c134b2a	latest	In use	28 days ago	77.85 MB	[Play] [More] [Delete]

Now I want to run the container and be able to interact with it in the console. I'm the client, and I interact with the daemon:

```
blxucreep@DESKTOP-0FKDJG2:~$ docker run -it ubuntu:latest
root@42ce3485197e:/#
```

We can verify this in Docker directly, and some actions are available (run/stop etc...):



The screenshot shows the 'Containers' tab in Docker Desktop. It displays a table with one container, 'frosty\_bose', which is 'Running'. The container has a CPU usage of 0% and a memory usage of 952KB. The interface includes a search bar, a filter icon, and a 'Show charts' dropdown menu.

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
frosty_bose 42ce3485197	ubuntu:latest	Running	0%		3 minutes ago	[Stop] [More] [Delete]

Let's try some commands to see if our container works correctly. Here we see all the directories we have in our container:

```
root@42ce3485197e:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root
run  sbin  srv  sys  tmp   usr  var
root@42ce3485197e:/#
```

I can install some packages, like 'cowsay':

```
root@42ce3485197e:/# apt-get install cowsay
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```