

Doc Job11

created : 24-06-02 21:32

modified : 24-06-02 21:32

tags : [note]

References :

Nous allons maintenant réaliser l'entièreté du sujet une deuxième fois, mais cette fois-ci en GUI avec Portainer

Job 11.1)

<https://www.webhi.com/how-to/fr/installer-utiliser-portainer-pour-la-gestion-des-conteneurs-docker/>

```
docker run -d -p 9000:9000 --name portainer --restart always  
-v /var/run/docker.sock:/var/run/docker.sock -v  
portainer_data:/data portainer/portainer-ce
```

Voici ce que fait cette commande :

- `-d` : Exécute le conteneur en mode détaché.
- `-p 9000:9000` : Mappe le port 9000 sur votre hôte au port 9000 dans le conteneur.
- `--name portainer` : Nomme le conteneur "portainer".
- `--restart always` : Configure le conteneur pour redémarrer automatiquement.

- `-v /var/run/docker.sock:/var/run/docker.sock` : Donne à Portainer l'accès au socket Docker.
- `-v portainer_data:/data` : Persiste les données de Portainer dans un volume nommé "portainer_data".
- `portainer/portainer-ce` : Spécifie l'image Portainer Community Edition.

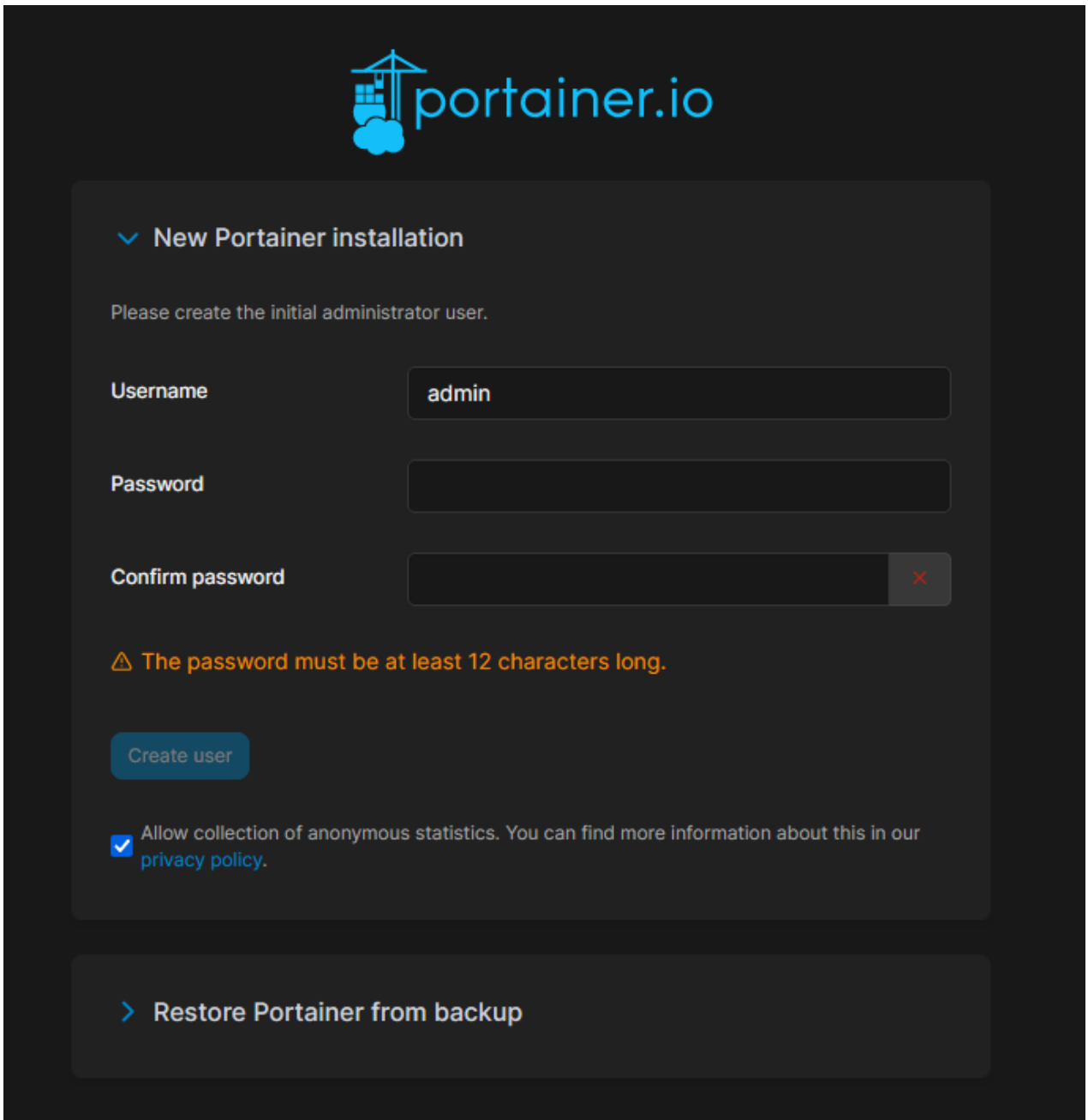
On accède ensuite à l'interface web:

<http://192.168.70.132:9000>

****Création du compte admin**

1. Cliquez sur le bouton "Créer un compte administrateur".
2. Entrez un nom d'utilisateur, un mot de passe et une adresse e-mail.

3. Cliquez sur “Créer l'utilisateur”.



The screenshot shows the Portainer.io web interface for creating a new installation. At the top is the Portainer.io logo. Below it, a section titled "New Portainer installation" with a dropdown arrow. Under this section, a message says "Please create the initial administrator user." There are three input fields: "Username" with the value "admin", "Password" (empty), and "Confirm password" (empty). A red 'x' icon is visible next to the "Confirm password" field. Below the fields, a warning message in orange text says "The password must be at least 12 characters long." Below the warning is a blue button labeled "Create user". At the bottom of the form, there is a checkbox that is checked, with the text "Allow collection of anonymous statistics. You can find more information about this in our [privacy policy](#)." Below the form, there is a button with a right-pointing arrow and the text "Restore Portainer from backup".

portainer.io

▼ New Portainer installation

Please create the initial administrator user.

Username

Password

Confirm password

⚠ The password must be at least 12 characters long.

Create user

☒ Allow collection of anonymous statistics. You can find more information about this in our [privacy policy](#).

> Restore Portainer from backup

login : admin

mdp : 123456789101112

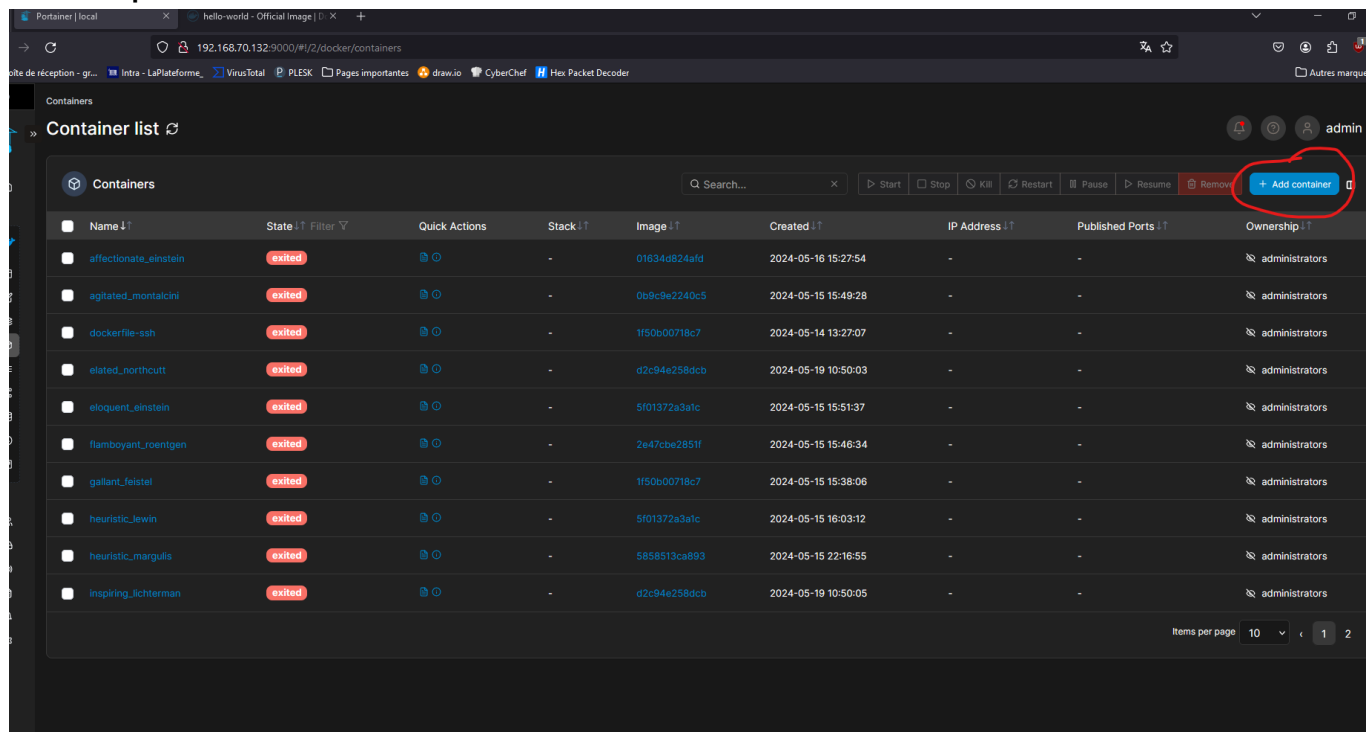
Voici la liste de mes images docker au moment T:

```
user@debiandocker:~$ sudo docker mages
docker: 'mages' is not a docker command.
See 'docker --help'
user@debiandocker:~$ sudo docker images
REPOSITORY              TAG               IMAGE ID           CREATED            SIZE
192.168.70.132:5000/web  latest           01634d824afd      2 days ago        48.3MB
<none>                  <none>           5858513ca893      3 days ago        185MB
nginx                   latest           e784f4560448      2 weeks ago       188MB
nginx                   alpine           501d84f5d064      2 weeks ago       48.3MB
portainer/portainer-ce  latest           a3f85c245ec3      3 weeks ago       293MB
registry                latest           d6b2c32a0f14      7 months ago      25.4MB
user@debiandocker:~$
```

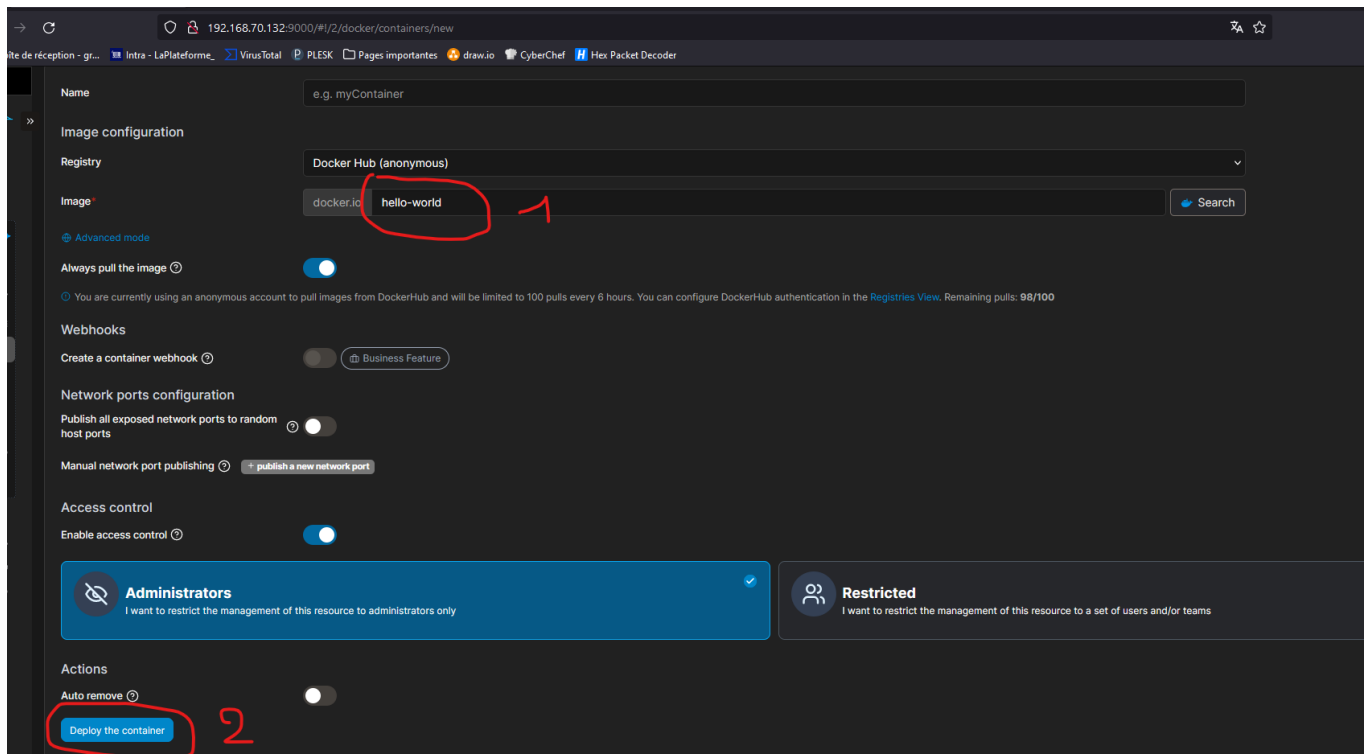
Job 11.2)

Nous allons donc télécharger le conteneur hello-world sur le registry officiel de Docker:

On clique sur "Add container":



On entre ensuite le nom du conteneur a importer, puis on clique sur "Deploy the container":



On peut voir en retournant sur notre machine que l'image s'est bien installée:

```
user@debiandocker:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
192.168.70.132:5000/web latest             01634d824afd       2 days ago         48.3MB
<none>              <none>             5858513ca893       3 days ago         185MB
nginx               latest             e784f4560448       2 weeks ago        188MB
nginx               alpine             501d84f5d064       2 weeks ago        48.3MB
portainer/portainer-ce latest             a3f85c245ec3       3 weeks ago        293MB
registry            latest             d6b2c32a0f14       7 months ago       25.4MB
hello-world         latest             d2c94e258dcb       12 months ago      13.3kB
user@debiandocker:~$
```

On démarre le conteneur pour voir s'il fonctionne bien :

Soit par l'onglet "Container", soit en CLI comme ci-dessous

```
hello-world      latest      d2c94e258dcb  12 months ago  13.3kB
user@debiandocker:~$ sudo docker run d2c94e258dcb

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

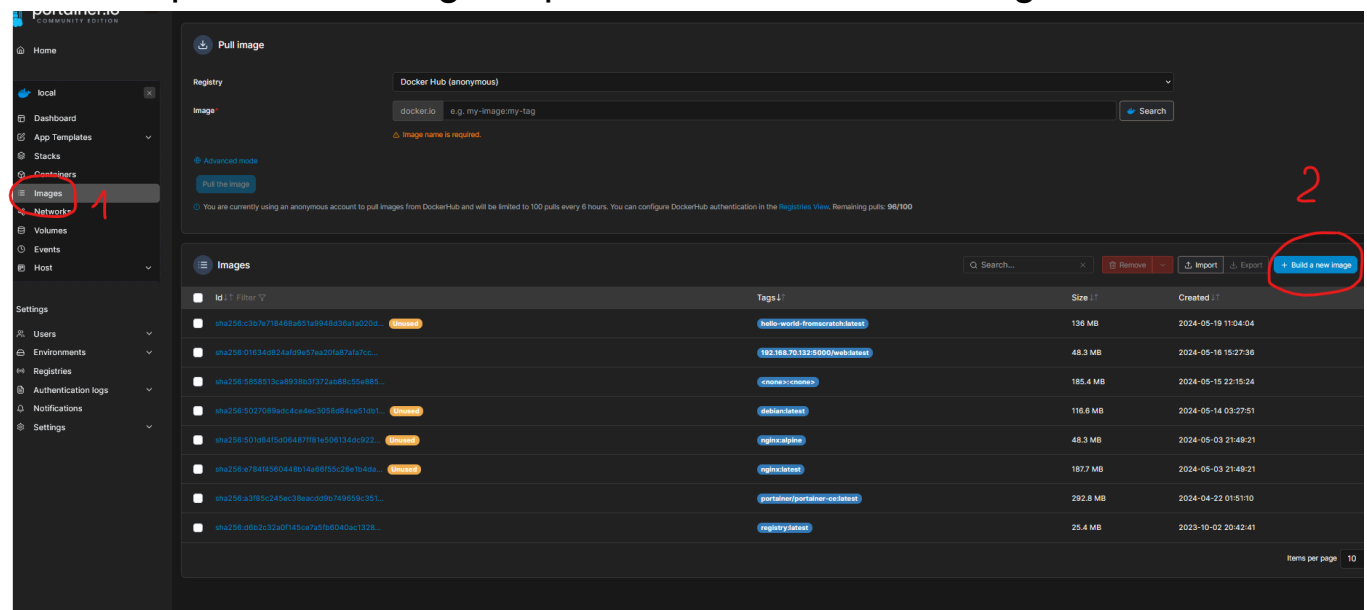
user@debiandocker:~$
```

Tada!

Job 11.3)

Nous devons maintenant déployer nous même un conteneur qui nous affichera "hello world". Pour se faire:

Nous cliquons sur "Images", puis sur "Build a new image":



On remplit ensuite le titre de notre image, son contenu (à la manière d'un dockerfile), puis on l'upload:

Names [add additional name](#)

A name must be specified in one of the following formats: `name:tag`, `repository/name:tag` or `registryfqdn:port/repository/name:tag` format. If you omit the tag the default `latest` value is assumed.

name

Build method

- Web editor** Use our Web editor
- Upload Upload from your computer
- URL Specify a URL to a file

Web editor

You can get more information about Dockerfile format in the [official documentation](#).

Define or paste the content of your Dockerfile here

```
1 FROM debian:latest
2
3 LABEL maintainer="user"
4
5 RUN apt-get update && apt-get install -y
6
7 ENTRYPOINT ["echo", "hello world"]
```

Upload

You can upload files from your local computer for referencing in your Dockerfile (using ADD filename) so they are included in your built image.

[Select files](#)

Actions

[Build the image](#)

On va vérifier sur notre machine hôte si l'image s'est créée, puis on la lance pour voir si elle fonctionne:

```
user@debiandocker:~$ sudo docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
hello-world-fromscratch latest      c3b7e718468a   7 seconds ago  136MB
192.168.70.132:5000/web latest      01634d824afd   2 days ago    48.3MB
<none>              <none>       5858513ca893   3 days ago    185MB
debian              latest      5027089adc4c   5 days ago    117MB
nginx               alpine      501d84f5d064   2 weeks ago   48.3MB
nginx               latest      e784f4560448   2 weeks ago   188MB
portainer/portainer-ce latest      a3f85c245ec3   3 weeks ago   293MB
registry            latest      d6b2c32a0f14   7 months ago  25.4MB
user@debiandocker:~$
```

```
user@debiandocker:~$ sudo docker run hello-world-fromscratch
hello world
user@debiandocker:~$
```

Tada!

Job 11.4)

Pour ce job, je vais réutiliser la méthode au dessus (Job 11.3) pour créer un dockerfile.

Je réutilise le code du Job 4 pour créer mon conteneur SSH.

Je vérifie s'il s'est bien créé, puis je le lance:

```
user@debiandocker:~$ sudo docker images
REPOSITORY          TAG          IMAGE ID      CREATED       SIZE
ssh-portainer        latest       2bf8a4db96bb 35 seconds ago 185MB
hello-world-fromscratch latest       c3b7e718468a 5 minutes ago 136MB
192.168.70.132:5000/web latest       01634d824afd 2 days ago    48.3MB
<none>              <none>      5858513ca893 3 days ago    185MB
debian              bookworm    5027089adc4c 5 days ago    117MB
debian              latest      5027089adc4c 5 days ago    117MB
nginx               latest      e784f4560448 2 weeks ago   188MB
nginx               alpine      501d84f5d064 2 weeks ago   48.3MB
portainer/portainer-ce latest      a3f85c245ec3 3 weeks ago   293MB
registry            latest      d6b2c32a0f14 7 months ago  25.4MB
user@debiandocker:~$ sudo docker run -d -p 2222:22 ssh-portainer
82a57f2d049488f0b6c9497712a419bf07463e9c4a21fffc3dc2958fa25ec9a2
user@debiandocker:~$
```

Je me connecte ensuite en SSH sur le conteneur qui est run:

```
user@debiandocker:~$ ssh root@localhost -p 2222
hostkeys_find_by_key hostfile: hostkeys_foreach failed for /home/user/.ssh/known_hosts: Permission denied
The authenticity of host '[localhost]:2222 ([::1]:2222)' can't be established.
ED25519 key fingerprint is SHA256:8uwiweZLWGD8E3o9jGWVdiDyqp6aVE6uEniDpmlGLa0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/home/user/.ssh/known_hosts).
root@localhost's password:
Permission denied, please try again.
root@localhost's password:
client_input_hostkeys: hostkeys_foreach failed for /home/user/.ssh/known_hosts: Permission denied
Linux 82a57f2d0494 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@82a57f2d0494:~#
```

Tada!

Job 11.5)

Je pense que le Job 5 est infaisable sur Portainer, étant donné que c'est une opération à réaliser sur notre VM par Bash. Je vais donc sauter ce Job.

Job 11.6)

Job contenant de la documentation que nous devons faire, je saute donc ce Job.

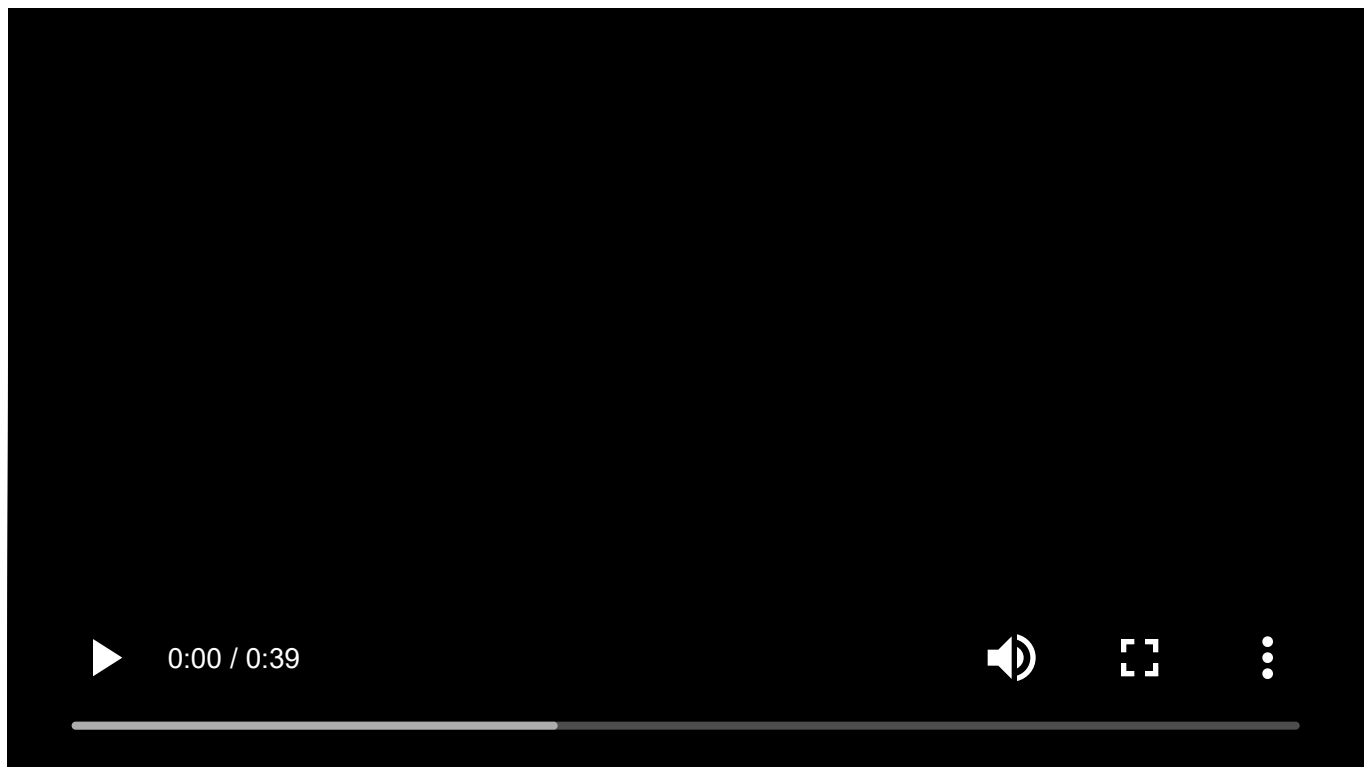
Job 11.7)

Pour créer un docker-compose, on doit utiliser la partie "Stacks".

On nomme notre Stack/yml, puis on envoie le code que l'on a créé dans le Job 07. Ensuite, on "Deploy the stack":

The screenshot shows the Portainer.io interface. On the left sidebar, the 'Stacks' menu item is highlighted with a red circle and the number 1. In the main area, the 'job7' stack is selected, with its name circled in red and labeled with a red 2. The 'Web editor' tab is active, showing a Docker Compose file with French comments. A large red circle labeled with a red 3 encompasses the entire Compose file content. At the bottom, the 'Actions' section contains a 'Deploy the stack' button, which is circled in red and labeled with a red 4. The interface also shows options for 'Build method' (Web editor, Upload, Repository) and 'Access control' (Administrators, Restricted).

Puis on réalise l'envoi de l' `index.html` par FTP:



Tada!

Job 11.8)

Comme dans les Job précédents, je créé une image, où j'y renvoie le code du **Job 08**:

Builder

Output

Naming

You can specify multiple names to your image.

Names [+ add additional name](#)

A name must be specified in one of the following formats: `name:tag`, `repository/name:tag` or `registryfqdn:port/repository/name:tag` format. If you omit the tag the default latest value is assumed.

☒

Build method

Web editor
Use our Web editor

Upload
Upload from your computer

URL
Specify a URL to a file

Web editor

You can get more information about Dockerfile format in the [official documentation](#).

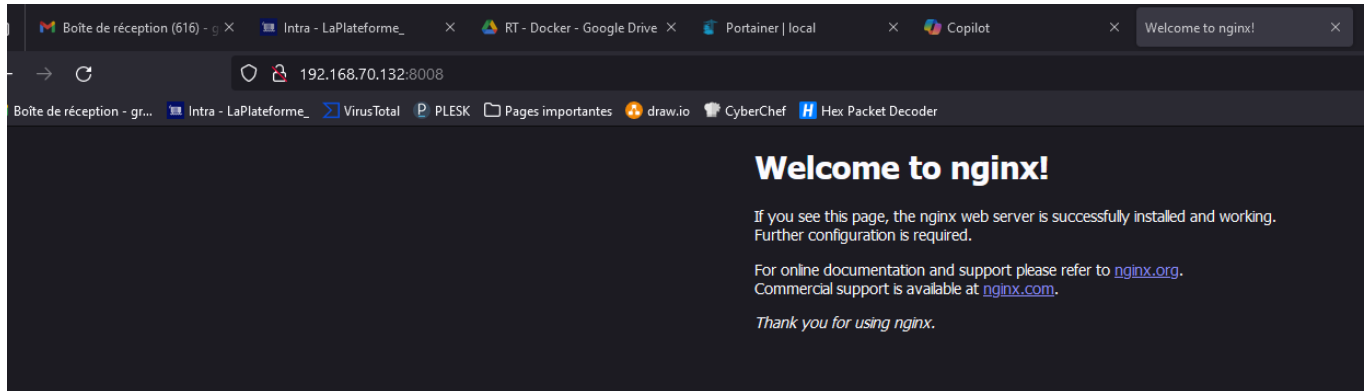
[🔗 Define or paste the content of your Dockerfile here](#) [🔗 Copy to clipboard](#)

```
1 # Base Image
2 FROM nginx:alpine
3
4 # MAINTAINER of the Dockerfile
5 MAINTAINER Greg <greg.dejoux@laplateforme.io>
6
7 #Copy the index.html file /usr/share/nginx/html/
8 COPY index.html /usr/share/nginx/html/
9
10 #Expose Nginx Port
11 EXPOSE 80
12
13 #Start NginxService.
14 #Le paramètre "-g" permet de définir des directives globales.
15 #Ici, "daemon off;" indique à Nginx de s'exécuter en premier plan, ce qui est nécessaire dans un conteneur Docker. En effet, si Nginx était lancé en tant que démon (c'est-à-dire en arrière-plan), le conteneur Docker se t
16 CMD ["nginx", "-g", "daemon off;"]
```

Je run ensuite le conteneur avec la commande qui suit :

```
sudo docker run -d -p 8008:80 web
```

et Tada!



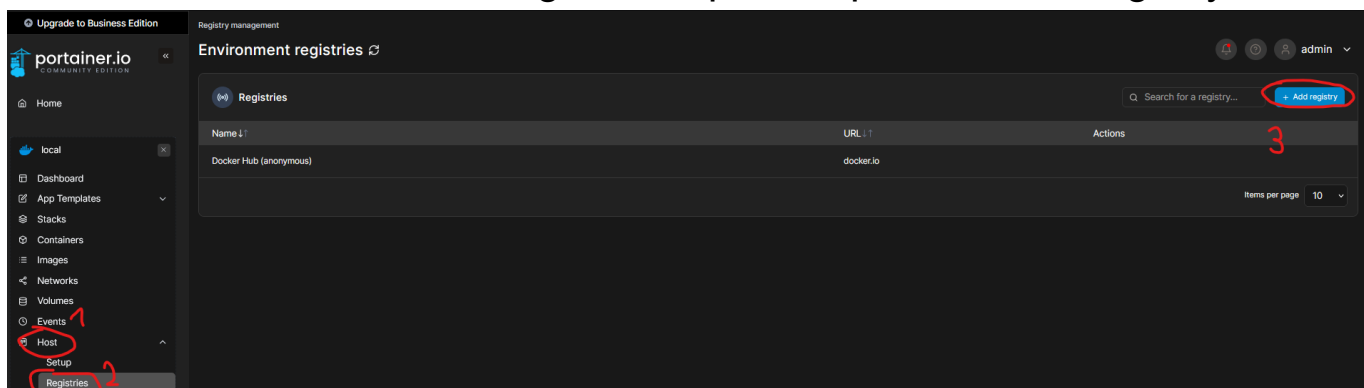
Job 11.9)

Tout d'abordn je clique sur mon compte en haut a de droite, puis dans ses settings.

Je génère un token

ptr_vtle+6fr44DyN7LzxM8aqR6DK5CbvVDKLp10NlnedEo=

Je me rend dans "Home>Registries", puis clique sur "Add registry":



Je la créé ensuite:

==Je stop le job 11.9 car je n'arrivais pas à mettre en place l'API