

Version - **2024.01**

Dernière mise-à-jour : 2024/04/01 11:16

DOF310 - Harbor

Contenu du Module

- Harbor
 - LAB #5 - Installation et Configuration de Harbor
 - 5.1 - Installation de Docker et de Docker Compose
 - 5.2 - Installation de Harbor
 - 5.3 - Création d'un Nouveau Projet
 - 5.4 - Création d'un Dépôt
 - 5.5 - Pousser des Images Supplémentaires
 - 5.6 - Travailler avec des Artefacts
 - 5.7 - Copier un Artefact d'un Projet vers un autre Projet
 - 5.8 - Utilisation de Labels
 - 5.9 - Authentification et RBAC
 - 5.10 - Utilisation d'Images Docker Personnalisées
 - 5.11 - Rétention de Tags
 - 5.12 - Immutabilité des Tags

Ressources

Lab #5

- wget <https://www.dropbox.com/scl/fi/tww1xm3wtpgnv974k6vbr/myEntrypoint.sh?rlkey=qypry4ryhoax8ddtqzy5td16d&dl=0>
 - wget <https://www.dropbox.com/scl/fi/bu8z1a5dxjs22gfy9fugb/Dockerfile?rlkey=nnl68r193nc6lor6429n03322&dl=0>
-

Harbor

Harbor est un registre open source qui :

- sécurise les artefacts (images) avec des politiques
- utilise un contrôle d'accès basé sur les rôles
- garantit que les images sont analysées et exemptes de vulnérabilités
- signe les images comme fiables.

LAB #5 - Installation et Configuration de Harbor

5.1 - Installation de Docker et de Docker Compose

Connectez-vous à votre instance de Debian11 en tant que trainee au 10.0.2.46 puis devenez l'utilisateur **root**.

Installez les dépendances de Docker :

```
root@debian11:~# apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common
```

Téléchargez la clef GPG officielle de docker :

```
root@debian11:~# curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -  
OK
```

Ajoutez le dépôt stable de docker :

```
root@debian11:~# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs)  
stable"
```

Installez maintenant **docker-ce** :

```
root@debian11:~# apt-get install docker-ce lvm2 libdevmapper-dev
```

Téléchargez docker-compose :

```
root@debian11:~# wget https://github.com/docker/compose/releases/download/v2.14.0/docker-compose-linux-x86_64
```

Rendez docker-compose exécutable et déplacez-le dans le répertoire **/usr/bin** :

```
root@debian11:~# chmod +x docker-compose-linux-x86_64
```

```
root@debian11:~# mv docker-compose-linux-x86_64 /usr/bin/docker-compose
```

5.2 - Installation de Harbor

Téléchargez Harbor :

```
root@debian11:~# wget  
https://github.com/goharbor/harbor/releases/download/v2.6.2/harbor-offline-installer-v2.6.2.tgz
```

Désarchivez Harbor :

```
root@debian11:~# tar xvf harbor-offline-installer*.tgz  
harbor/harbor.v2.6.2.tar.gz  
harbor/prepare  
harbor/LICENSE  
harbor/install.sh  
harbor/common.sh  
harbor/harbor.yml.tpl
```

Placez-vous dans le répertoire **harbor** et copiez le fichier **harbor.yml.tpl** vers **harbor.yml** :

```
root@debian11:~# cd harbor/
```

```
root@debian11:~/harbor# cp harbor.yml.tpl harbor.yml
```

Créez le fichier **/etc/docker/daemon.json** afin de configurer Docker pour l'utilisation du dépôt non-sécurisé :

```
root@debian11:~/harbor# vi /etc/docker/daemon.json
root@debian11:~/harbor# cat /etc/docker/daemon.json
{
"insecure-registries" : ["10.0.2.46:5000", "0.0.0.0"]
}
```

Modifiez maintenant l'entrée **hostname** du fichier **harbor/harbor.yml** et commentez la section concernant les connexions https :

```
root@debian11:~/harbor# vi harbor.yml
root@debian11:~/harbor# head -n 20 harbor.yml
# Configuration file of Harbor

# The IP address or hostname to access admin UI and registry service.
# DO NOT use localhost or 127.0.0.1, because Harbor needs to be accessed by external clients.
hostname: 10.0.2.46

# http related config
http:
  # port for http, default is 80. If https enabled, this port will redirect to https port
  port: 80

# https related config
# https:
# https port for harbor, default is 443
# port: 443
# The path of cert and key files for nginx
# certificate: /your/certificate/path
# private_key: /your/private/key/path
```

```
# # Uncomment following will enable tls communication between all harbor components
```

Constatez la présence du script **install.sh** dans le répertoire **harbor/** :

```
root@debian11:~/harbor# ls
common.sh          harbor.yml          install.sh  prepare
harbor.v2.6.2.tar.gz  harbor.yml.tpl  LICENSE
```

Désactivez le service nginx dans la VM Debian11 :

```
root@debian11:~/harbor# systemctl stop nginx
root@debian11:~/harbor# systemctl disable nginx
```

Installez maintenant Harbor avec le support pour **Chart Museum** et **Trivy** :

```
root@debian11:~/harbor# ./install.sh --with-trivy --with-chartmuseum
[Step 0]: checking if docker is installed ...
Note: docker version: 20.10.21
[Step 1]: checking docker-compose is installed ...
Note: docker-compose version: 2.14.0
[Step 2]: loading Harbor images ...
...
[Step 3]: preparing environment ...
...
[Step 4]: preparing harbor configs ...
...
[Step 5]: starting Harbor ...
→
```

Chartmuseum will be deprecated as of Harbor v2.6.0 and start to be removed in v2.8.0 or later.

Please see discussion here for more details. <https://github.com/goharbor/harbor/discussions/15057>

[+] Running 13/13

:: Network harbor_harbor Created

0.2s

:: Network harbor_harbor-chartmuseum Created

0.3s

:: Container harbor-log Started

14.1s

:: Container registry Started

15.8s

:: Container registryctl Started

18.1s

:: Container harbor-portal Started

18.1s

:: Container redis Started

16.9s

:: Container chartmuseum Started

15.7s

:: Container harbor-db Started

16.4s

:: Container trivy-adapter Started

14.8s

:: Container harbor-core Started

15.0s

:: Container nginx Started

13.4s

:: Container harbor-jobservice Started

13.2s

✓ ----Harbor has been installed and started successfully.----

Constatez les conteneurs créés par l'installation :

```
root@debian11:~/harbor# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
b8e4e3934c7a (healthy)	goharbor/harbor-jobservice:v2.6.2	"/harbor/entrypoint..." harbor-jobservice	15 minutes ago	Up 15 minutes
579b4b43877f (healthy)	goharbor/nginx-photon:v2.6.2 0.0.0.0:80->8080/tcp, :::80->8080/tcp	"nginx -g 'daemon of..." nginx	15 minutes ago	Up 15 minutes
57c8697b4d39 (healthy)	goharbor/trivy-adapter-photon:v2.6.2	"/home/scanner/entry..." trivy-adapter	16 minutes ago	Up 15 minutes
ce4823c62886 (healthy)	goharbor/harbor-core:v2.6.2	"/harbor/entrypoint..." harbor-core	16 minutes ago	Up 15 minutes
57992f8f573a (healthy)	goharbor/chartmuseum-photon:v2.6.2	"/docker-entrypoint..." chartmuseum	16 minutes ago	Up 15 minutes
e2df863f7009 (healthy)	goharbor/harbor-portal:v2.6.2	"nginx -g 'daemon of..." harbor-portal	16 minutes ago	Up 15 minutes
3e7e2093a4a2 (healthy)	goharbor/harbor-db:v2.6.2	"/docker-entrypoint..." harbor-db	16 minutes ago	Up 15 minutes
406aa384c742 (healthy)	goharbor/redis-photon:v2.6.2	"redis-server /etc/r..." redis	16 minutes ago	Up 15 minutes
ddf08dcc8390 (healthy)	goharbor/harbor-registryctl:v2.6.2	"/home/harbor/start..." registryctl	16 minutes ago	Up 15 minutes
b5cedbab739d (healthy)	goharbor/registry-photon:v2.6.2	"/home/harbor/entryp..." registry	16 minutes ago	Up 15 minutes
13665ace83fd (healthy)	goharbor/harbor-log:v2.6.2 127.0.0.1:1514->10514/tcp	"/bin/sh -c /usr/loc..." harbor-log	16 minutes ago	Up 15 minutes

Ainsi que les images téléchargées :

```
root@debian11:~/harbor# docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
goharbor/harbor-exporter  v2.6.2      ae230e335853     4 weeks ago     95.4MB
goharbor/chartmuseum-photon  v2.6.2      be361eee8a36     4 weeks ago     225MB
goharbor/redis-photon      v2.6.2      84fa3669bb10     4 weeks ago     154MB
goharbor/trivy-adapter-photon v2.6.2      beef2527c7b2     4 weeks ago     441MB
goharbor/notary-server-photon v2.6.2      b43e7c963f06     4 weeks ago     112MB
```

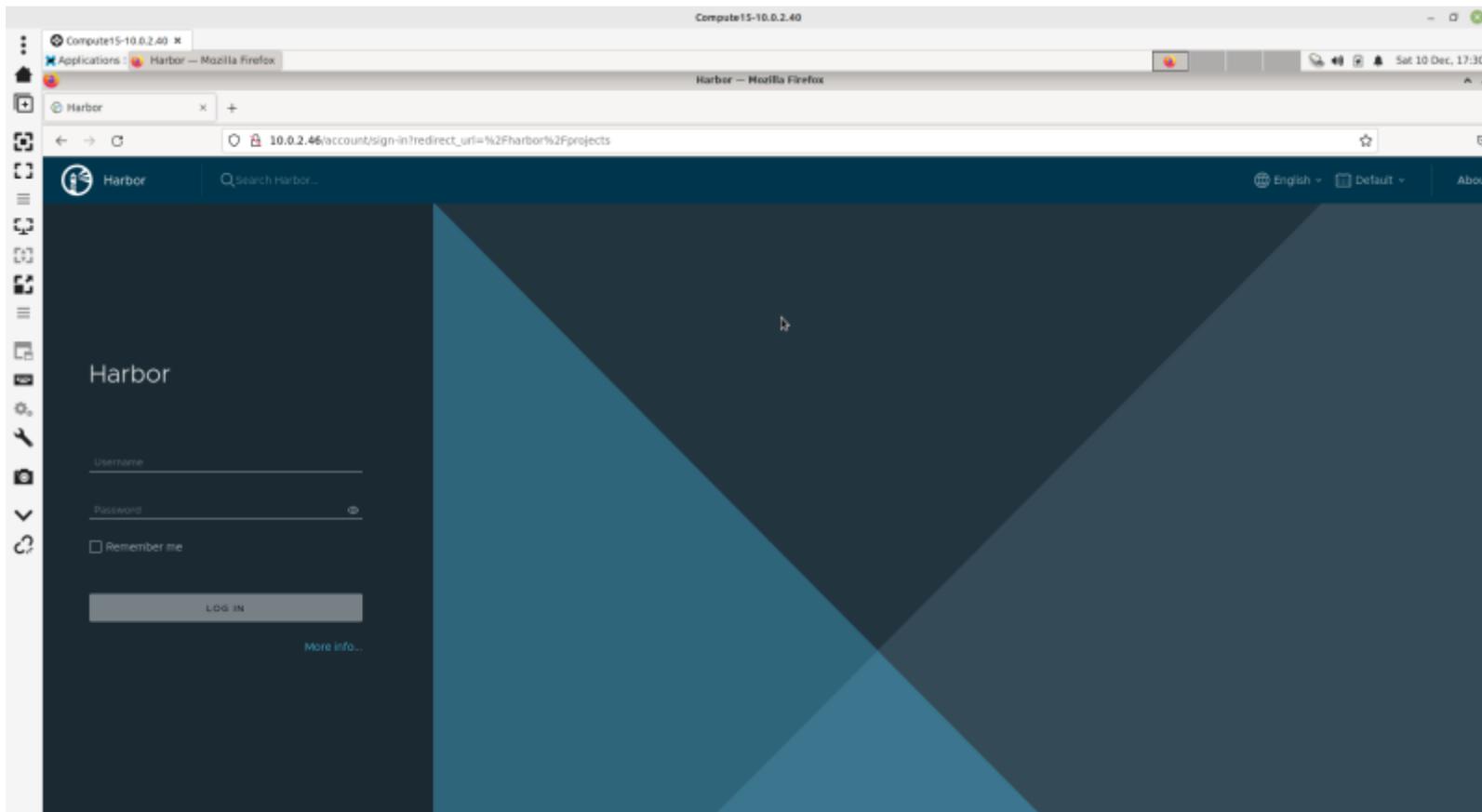
goharbor/notary-signer-photon	v2.6.2	4e9982286316	4 weeks ago	109MB
goharbor/harbor-registryctl	v2.6.2	d403d4e0f990	4 weeks ago	137MB
goharbor/registry-photon	v2.6.2	596e445b18f0	4 weeks ago	77.5MB
goharbor/nginx-photon	v2.6.2	cbbdf28947a5	4 weeks ago	154MB
goharbor/harbor-log	v2.6.2	3ce902023f73	4 weeks ago	161MB
goharbor/harbor-jobservice	v2.6.2	05791a75c8b8	4 weeks ago	250MB
goharbor/harbor-core	v2.6.2	484e2a64616d	4 weeks ago	213MB
goharbor/harbor-portal	v2.6.2	932297030d4e	4 weeks ago	162MB
goharbor/harbor-db	v2.6.2	351b19e366cb	4 weeks ago	225MB
goharbor/prepare	v2.6.2	eecc993c7643	4 weeks ago	164MB

Constatez maintenant les réseaux créés par l'installation :

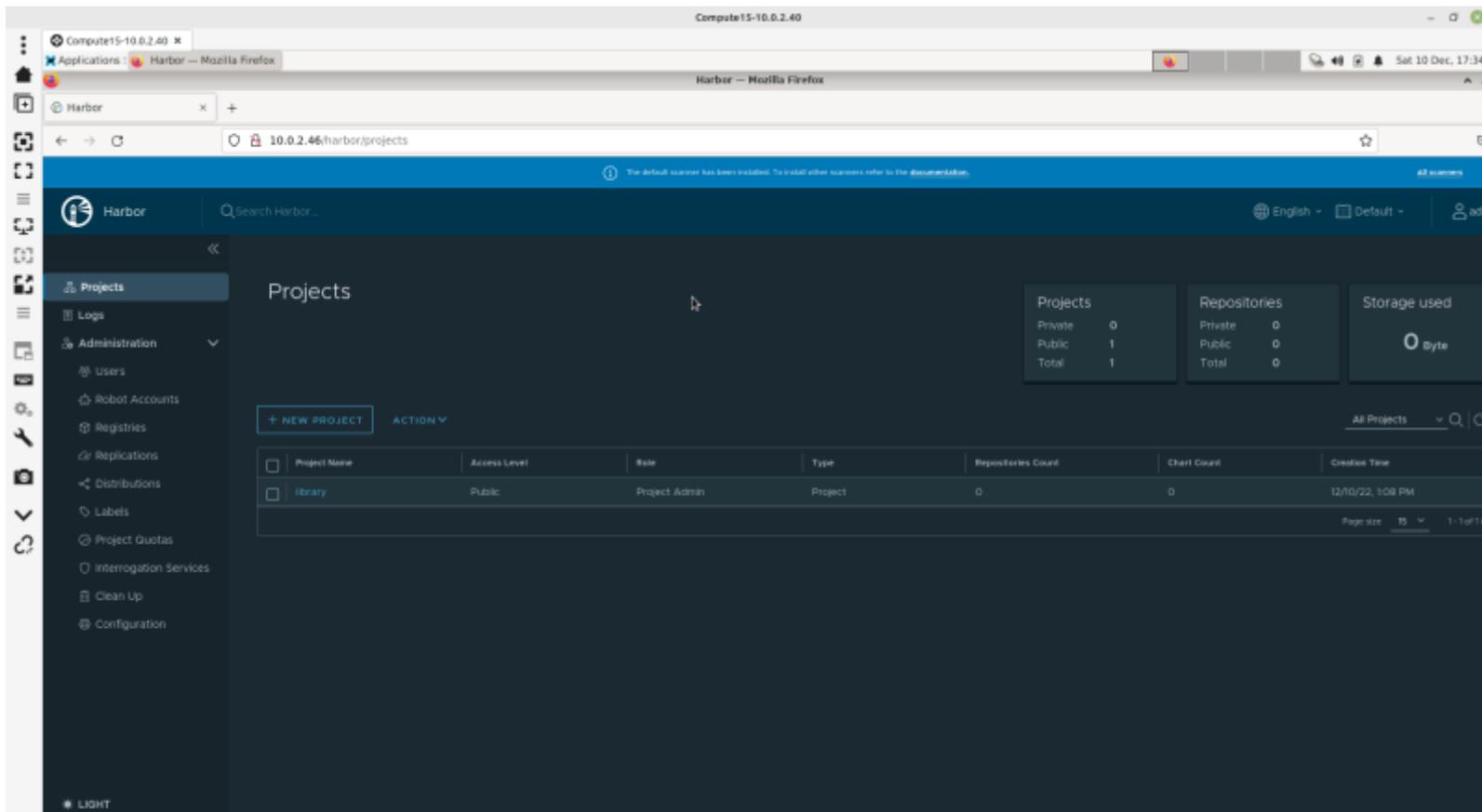
```
root@debian11:~/harbor# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
c080bf952007       bridge             bridge              local
380a2401fb01       harbor_harbor      bridge              local
b5d0cc4623f6       harbor_harbor-chartmuseum  bridge              local
3e48278ae275       host               host                local
4dd9174d8e25       none               null                local
```

5.3 - Création d'un Nouveau Projet

Connectez-vous à votre VM **Gateway_10.0.2.40_VNC** et ouvrez le navigateur web. Allez à l'URL <http://10.0.2.46> :



Connectez-vous avec les identifiants par défaut - admin / Harbor12345 :



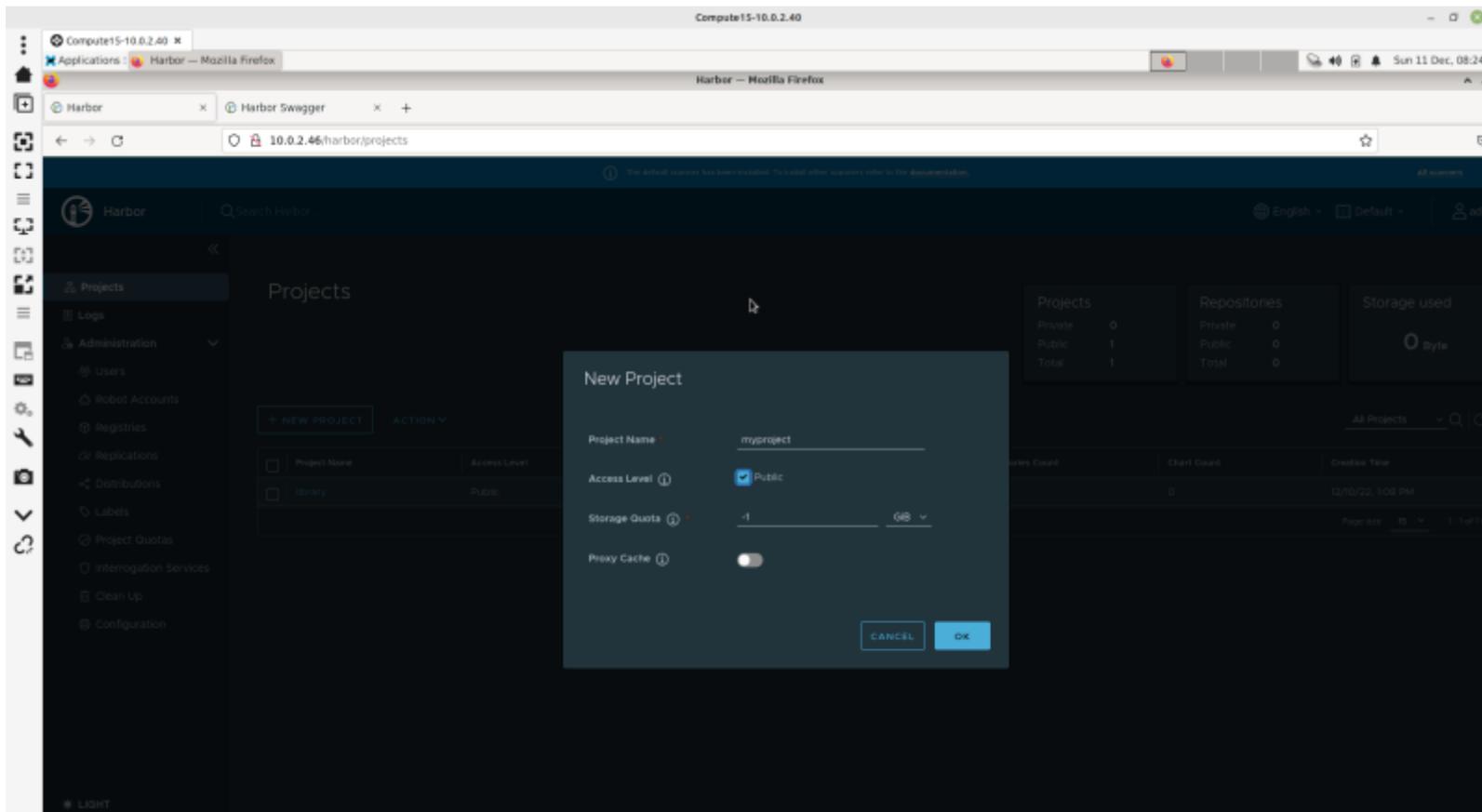
The screenshot shows the Harbor web interface in a browser window. The main content area is titled "Projects" and features a "+ NEW PROJECT" button and an "ACTION" dropdown menu. Below this is a table listing projects. The table has columns for Project Name, Access Level, Role, Type, Repositories Count, Chart Count, and Create Time. One project is listed: "library" with Access Level "Public", Role "Project Admin", Type "Project", Repositories Count "0", Chart Count "0", and Create Time "12/10/22, 1:00 PM".

<input type="checkbox"/>	Project Name	Access Level	Role	Type	Repositories Count	Chart Count	Create Time
<input type="checkbox"/>	library	Public	Project Admin	Project	0	0	12/10/22, 1:00 PM

Summary statistics on the right side of the page:

Projects		Repositories		Storage used
Private	0	Private	0	0 Byte
Public	1	Public	0	
Total	1	Total	0	

Harbor est organisé autour de **Projets**. Dans un projet se trouvent des **Dépôts** ou *Repositories*. Le projet nous permet de configurer le RBAC. Créez donc un nouveau projet en cliquant sur le bouton **New Project** :



The screenshot shows the Harbor web interface in a browser window. The main page is titled 'Projects' and features a sidebar with navigation options like 'Users', 'Robot Accounts', and 'Configurations'. A 'New Project' modal dialog is displayed in the center, with the following fields and values:

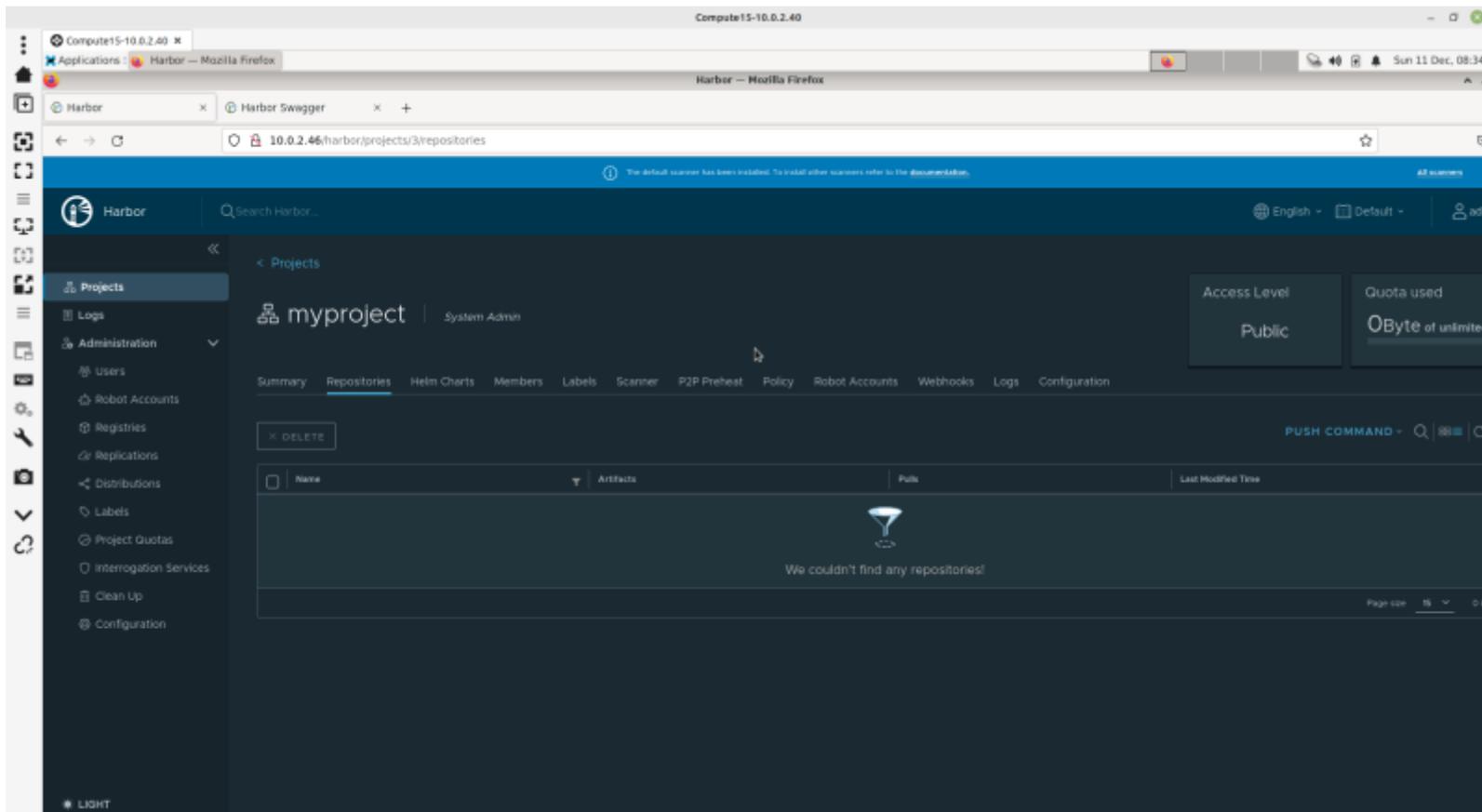
- Project Name: myproject
- Access Level: Public
- Storage Quota: -1 GB
- Proxy Cache:

The 'OK' button is highlighted in blue, indicating it is the next step in the process.



Important : Notez que la valeur de **-1** indique que le projet a un quota de disque **illimité**.

Cliquez sur le bouton **OK** puis cliquez sur le nom **myproject** :



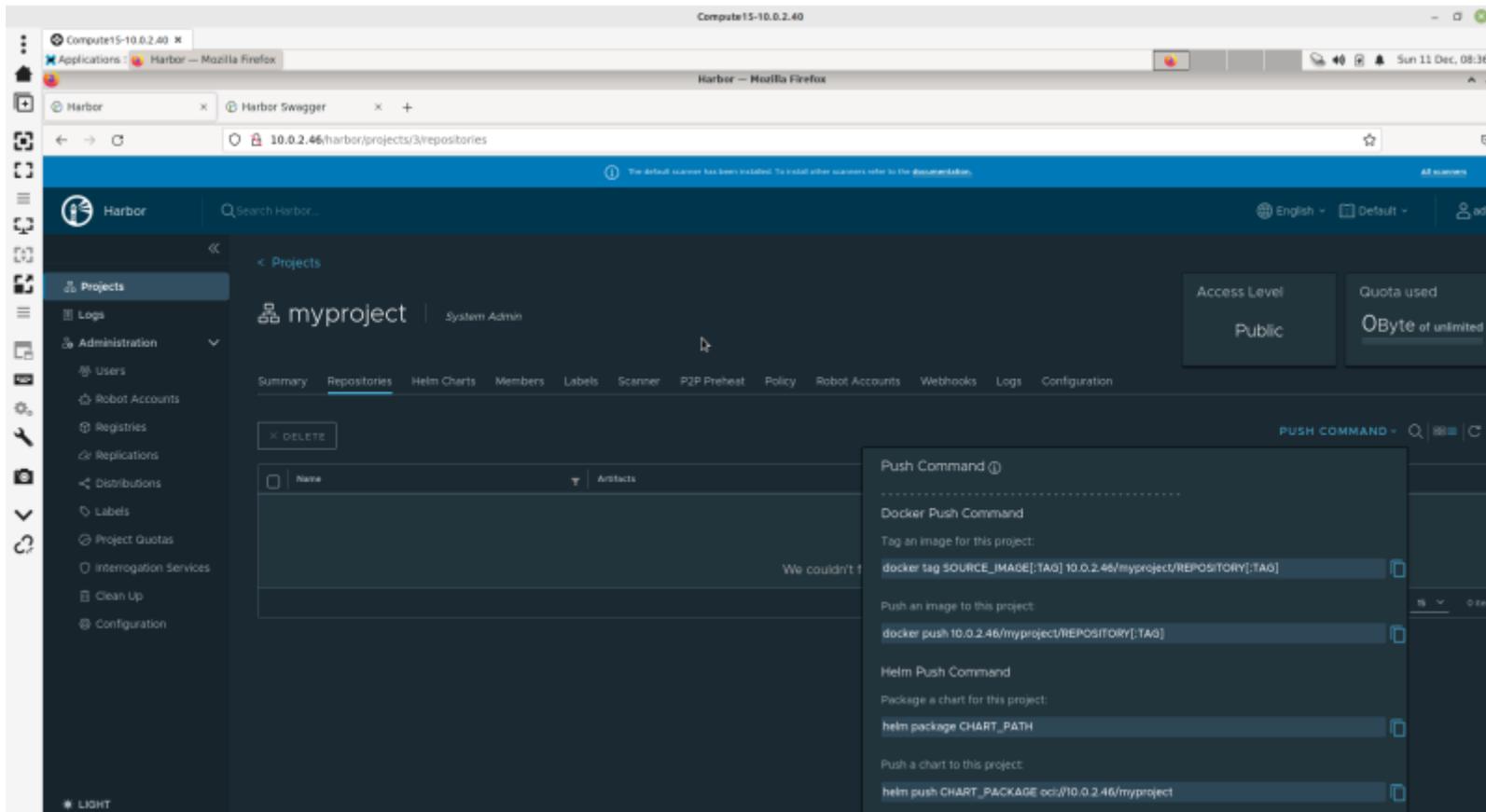
5.4 - Création d'un Dépôt

Téléchargez maintenant l'image de **hello-world** à partir de docker.io :

```
root@debian11:~/harbor# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest
```

```
docker.io/library/hello-world:latest
```

Retournez dans Harbor et cliquez sur **PUSH COMMAND** à droite. Dans la liste des commandes vous vous apercevrez la commande pour *tagger* une image pour le projet **myproject** :



Copiez cette commande puis modifiez la ainsi :

```
root@debian11:~/harbor# docker tag hello-world 10.0.2.46:80/myproject/hello-world
```

```
root@debian11:~/harbor# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
goharbor/harbor-exporter	v2.6.2	ae230e335853	4 weeks ago	95.4MB
goharbor/chartmuseum-photon	v2.6.2	be361eee8a36	4 weeks ago	225MB
goharbor/redis-photon	v2.6.2	84fa3669bb10	4 weeks ago	154MB
goharbor/trivy-adapter-photon	v2.6.2	beef2527c7b2	4 weeks ago	441MB
goharbor/notary-server-photon	v2.6.2	b43e7c963f06	4 weeks ago	112MB
goharbor/notary-signer-photon	v2.6.2	4e9982286316	4 weeks ago	109MB
goharbor/harbor-registryctl	v2.6.2	d403d4e0f990	4 weeks ago	137MB
goharbor/registry-photon	v2.6.2	596e445b18f0	4 weeks ago	77.5MB
goharbor/nginx-photon	v2.6.2	cbbdf28947a5	4 weeks ago	154MB
goharbor/harbor-log	v2.6.2	3ce902023f73	4 weeks ago	161MB
goharbor/harbor-jobservice	v2.6.2	05791a75c8b8	4 weeks ago	250MB
goharbor/harbor-core	v2.6.2	484e2a64616d	4 weeks ago	213MB
goharbor/harbor-portal	v2.6.2	932297030d4e	4 weeks ago	162MB
goharbor/harbor-db	v2.6.2	351b19e366cb	4 weeks ago	225MB
goharbor/prepare	v2.6.2	eecc993c7643	4 weeks ago	164MB
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB
10.0.2.46:80/myproject/hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB



Important : Notez la spécification du port **80**. Sans spécifier le port le port par défaut **443** serait utilisé.

Connectez-vous à Harbor :

```
root@debian11:~/harbor# docker login 10.0.2.46:80
Username: admin
Password: Harbor12345
Error response from daemon: Get "https://10.0.2.46:80/v2/": http: server gave HTTP response to HTTPS client
```



Important : Notez l'erreur **Error response from daemon: Get**



“https://10.0.2.46:80/v2/” : http: server gave HTTP response to HTTPS client. Ceci est du à la configuration de docker dans le fichier `/etc/docker/daemon.json` où le port 80 n'a pas été spécifié :

```
root@debian11:~/harbor# cat /etc/docker/daemon.json
{
"insecure-registries" : ["10.0.2.46:5000", "0.0.0.0"]
}
```

Modifiez donc ce fichier :

```
root@debian11:~/harbor# vi /etc/docker/daemon.json
root@debian11:~/harbor# cat /etc/docker/daemon.json
{
"insecure-registries" : ["10.0.2.46:5000", "10.0.2.46:80", "0.0.0.0"]
}
```

Redémarrez le service docker :

```
root@debian11:~/harbor# systemctl restart docker
root@debian11:~/harbor# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-12-11 08:57:36 CET; 20s ago
 TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
  Main PID: 487567 (dockerd)
    Tasks: 44
   Memory: 56.2M
     CPU: 2.275s
   CGroup: /system.slice/docker.service
           └─487567 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

```
|487820 /usr/bin/docker-proxy -proto tcp -host-ip 127.0.0.1 -host-port 1514 -contain>  
|488485 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 80 -container-i>  
|488494 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 80 -container-ip 172>
```

```
Dec 11 08:57:14 debian11 dockerd[487567]: time="2022-12-11T08:57:14.186814789+01:00" level=info ms>  
Dec 11 08:57:14 debian11 dockerd[487567]: time="2022-12-11T08:57:14.571406314+01:00" level=info ms>  
Dec 11 08:57:19 debian11 dockerd[487567]: time="2022-12-11T08:57:19.175392656+01:00" level=info ms>  
Dec 11 08:57:21 debian11 dockerd[487567]: time="2022-12-11T08:57:21.464318214+01:00" level=error m>  
Dec 11 08:57:21 debian11 dockerd[487567]: time="2022-12-11T08:57:21.464360074+01:00" level=error m>  
Dec 11 08:57:31 debian11 dockerd[487567]: time="2022-12-11T08:57:31.545302044+01:00" level=info ms>  
Dec 11 08:57:35 debian11 dockerd[487567]: time="2022-12-11T08:57:35.749031343+01:00" level=info ms>  
Dec 11 08:57:36 debian11 dockerd[487567]: time="2022-12-11T08:57:36.596800068+01:00" level=info ms>
```

Maintenant redémarrez Harbor :

```
root@debian11:~/harbor# docker-compose down -v
```

```
[+] Running 13/13
```

```
  :: Container harbor-jobservice      Removed      5.7s  
  :: Container nginx                  Removed      5.8s  
  :: Container chartmuseum            Removed      0.2s  
  :: Container trivy-adapter           Removed      5.8s  
  :: Container registryctl            Removed      13.9s  
  :: Container harbor-portal           Removed      4.7s  
  :: Container harbor-core             Removed      5.3s  
  :: Container harbor-db               Removed      8.9s  
  :: Container registry                Removed      4.0s  
  :: Container redis                   Removed      8.7s  
  :: Container harbor-log              Removed      14.2s  
  :: Network harbor_harbor             Removed      0.8s  
  :: Network harbor_harbor-chartmuseum Removed      0.6s
```

```
root@debian11:~/harbor# docker-compose up -d
```

```
[+] Running 13/13
```

```
  :: Network harbor_harbor             Created      0.4s
```

```
⚡ Network harbor_harbor-chartmuseum Created 0.3s
⚡ Container harbor-log Started 13.5s
⚡ Container harbor-db Started 12.9s
⚡ Container harbor-portal Started 14.8s
⚡ Container chartmuseum Started 14.3s
⚡ Container redis Started 16.8s
⚡ Container registryctl Started 16.3s
⚡ Container registry Started 15.2s
⚡ Container trivy-adapter Started 14.0s
⚡ Container harbor-core Started 14.9s
⚡ Container harbor-jobservice Started 14.8s
⚡ Container nginx Started 15.0s
```

Connectez-vous à Harbor :

```
root@debian11:~/harbor# docker login 10.0.2.46:80
Username: admin
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Les coordonnées de connexion sont stockées dans le fichier **/root/.docker/config.json** :

```
root@debian11:~/harbor# cat /root/.docker/config.json
{
  "auths": {
    "10.0.2.46:80": {
      "auth": "YWRtaW46SGFyYm9yMTIzNDU="
    }
  }
}
```

```
}root@debian11:~/harbor#
```

De cette façon, lors des logins futur, docker ne demandera ni le nom de l'utilisateur, ni le mot de passe :

```
root@debian11:~/harbor# docker logout
Removing login credentials for https://index.docker.io/v1/

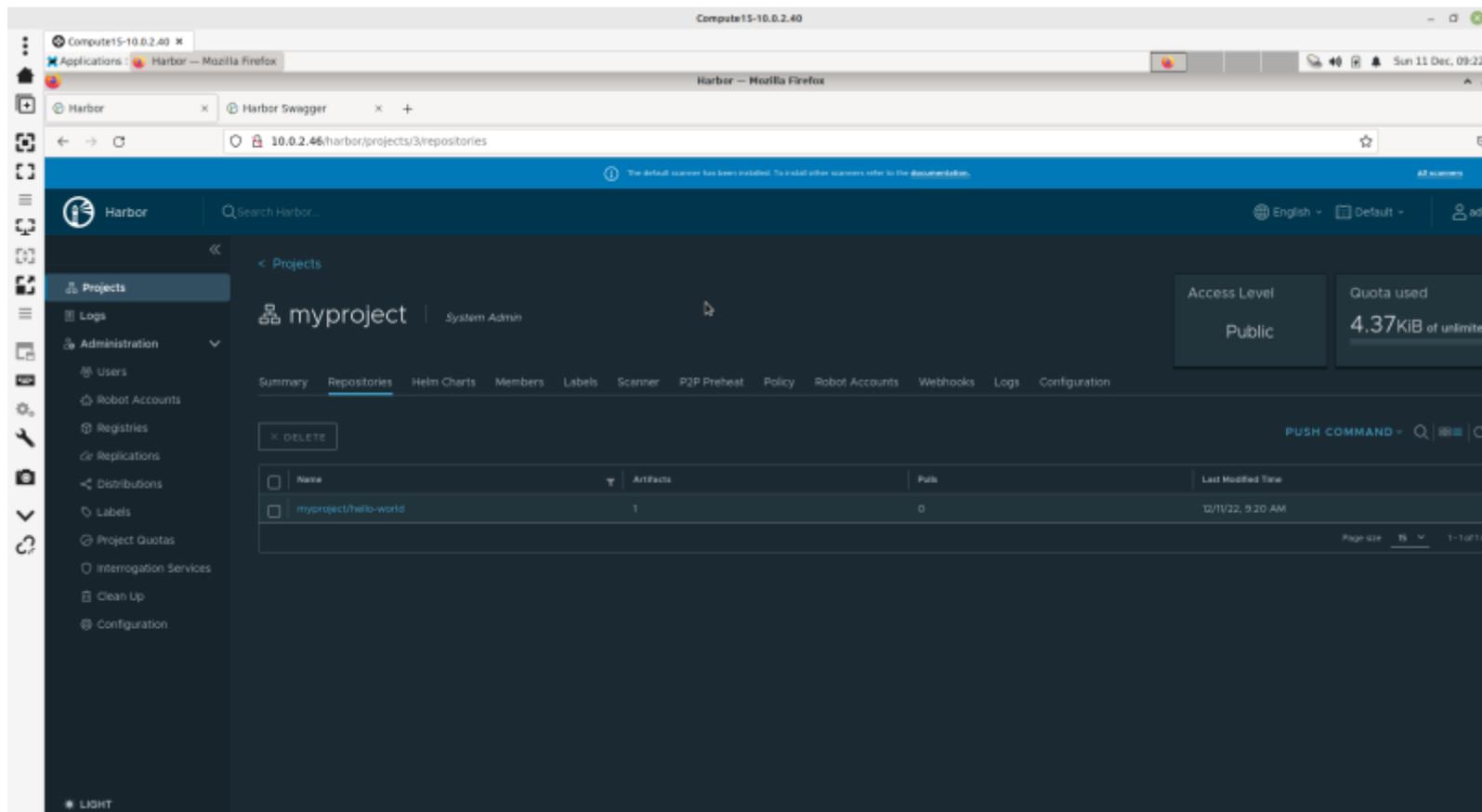
root@debian11:~/harbor# docker login 10.0.2.46:80
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Procédez maintenant au **push** :

```
root@debian11:~/harbor# docker push 10.0.2.46:80/myproject/hello-world
Using default tag: latest
The push refers to repository [10.0.2.46:80/myproject/hello-world]
e07ee1baac5f: Pushed
latest: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525
```

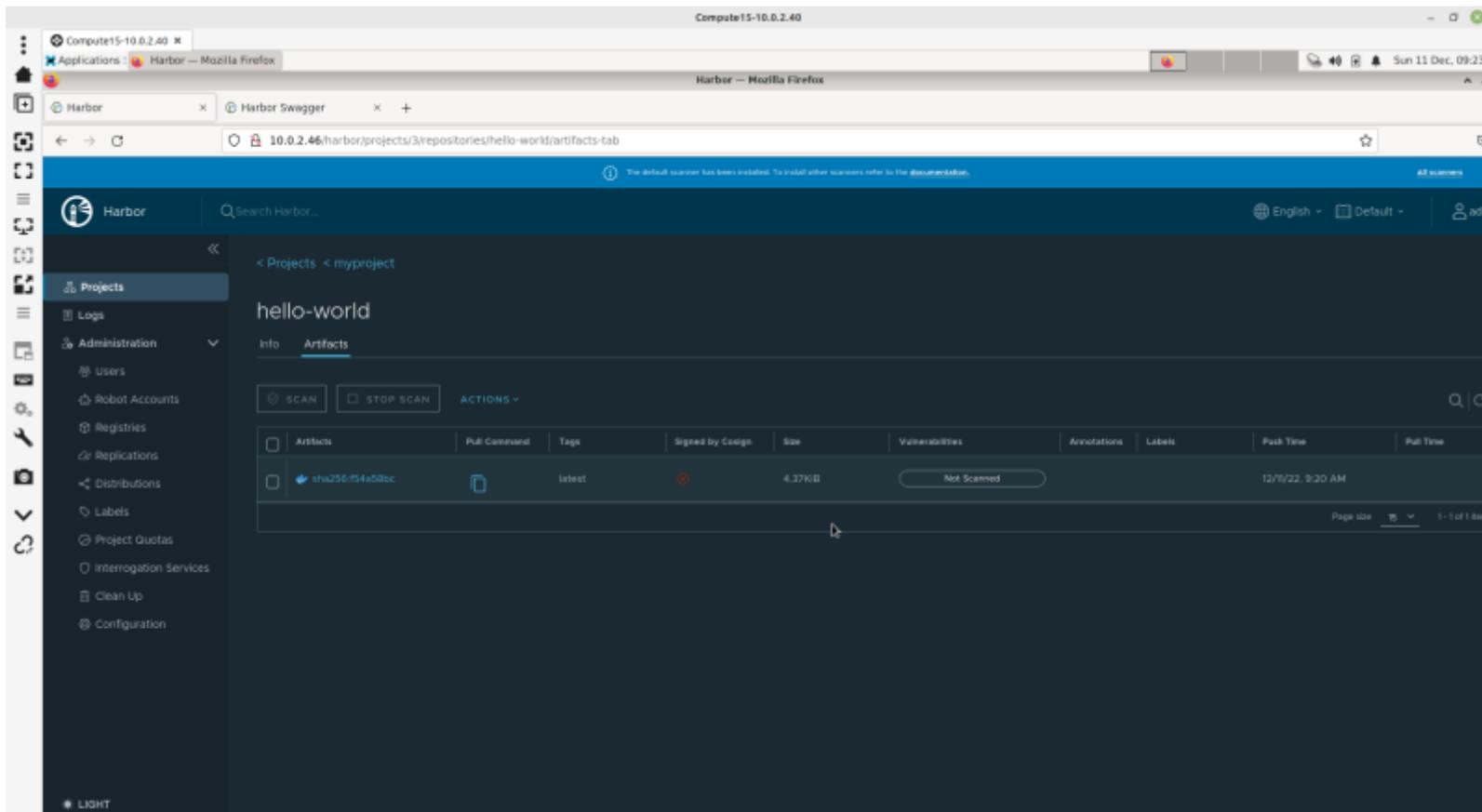
Retournez à Harbor et constatez la présence du dépôt :



The screenshot displays the Harbor web interface in a browser window. The address bar shows the URL `10.0.2.46/harbor/projects/3/repositories`. The interface is in French, with the user logged in as 'adm'. The main content area shows the 'Repositories' page for the project 'myproject', which is managed by 'System Admin'. The page includes a 'DELETE' button, a 'PUSH COMMAND' field, and a table of repositories. The table has columns for 'Name', 'Artifacts', 'Push', and 'Last Modified Time'. One repository is listed: 'myproject/hello-world' with 1 artifact and a last modified time of '12/11/22, 9:20 AM'. The page size is set to 15 items per page, showing 1 of 1 total items.

Name	Artifacts	Push	Last Modified Time
myproject/hello-world	1	0	12/11/22, 9:20 AM

Cliquez sur le nom du dépôt et constatez son contenu :



Important : Notez que l'image stocké dans Harbor est appelé un **Artefact**.

5.5 - Pousser des Images Supplémentaires

Modifiez le tag de l'image hello-world :

```
root@debian11:~/harbor# docker tag hello-world 10.0.2.46:80/myproject/hello-world:1.0
```

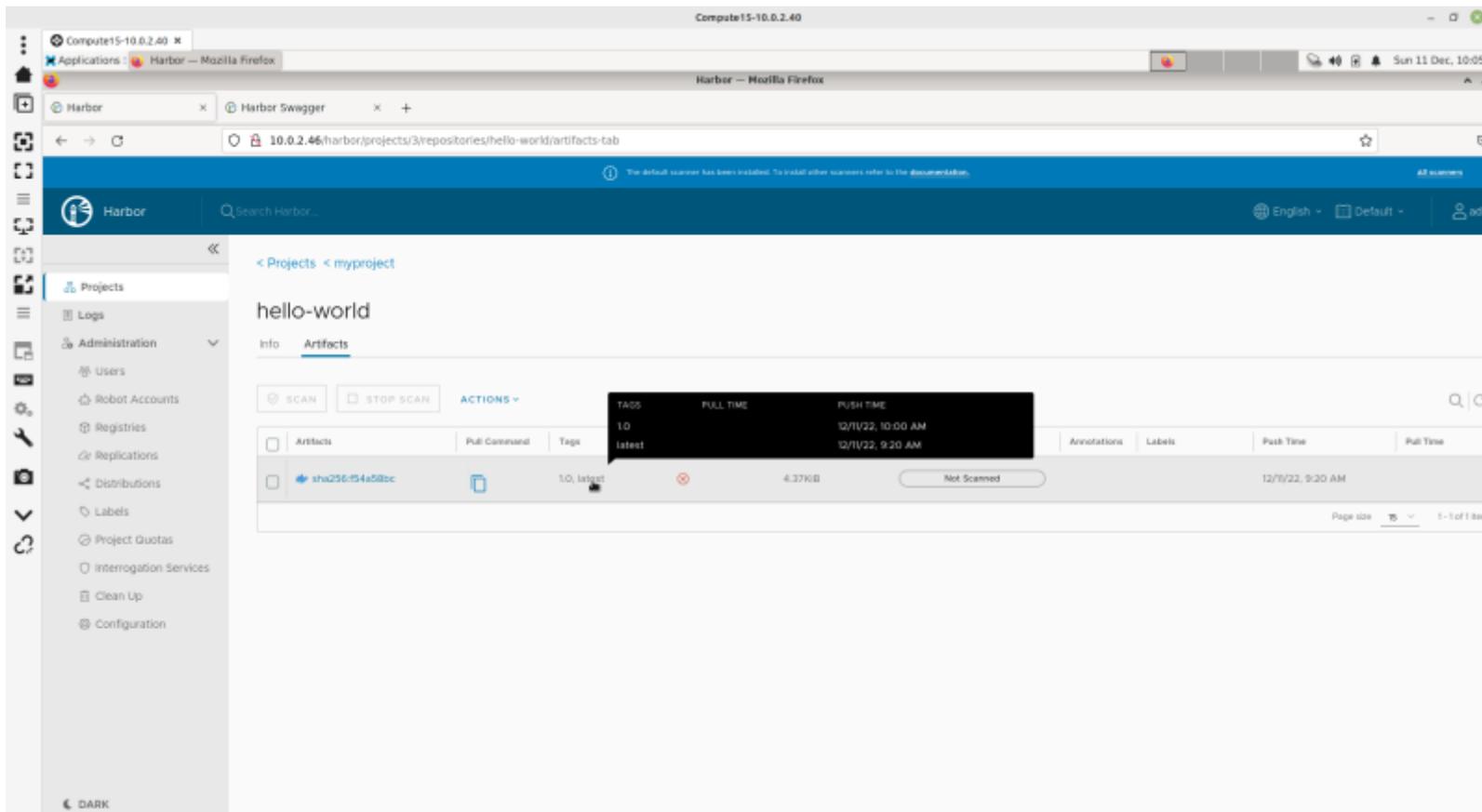
Procédez maintenant au **push** :

```
root@debian11:~/harbor# docker push 10.0.2.46:80/myproject/hello-world:1.0
The push refers to repository [10.0.2.46:80/myproject/hello-world]
e07eelbaac5f: Layer already exists
1.0: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525
```



Important : Notez que la valeur du SHA256 est indentique au push précédent. De cette façon Docker n'a pas poussé l'image mais seulement le nouveau tag.

Constatez le résultat dans Harbor :



Téléchargez maintenant l'image de **alpine** à partir de docker.io :

```
root@debian11:~/harbor# docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
c158987b0551: Pull complete
Digest: sha256:8914eb54f968791faf6a8638949e480fef81e697984fba772b3976835194c6d4
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

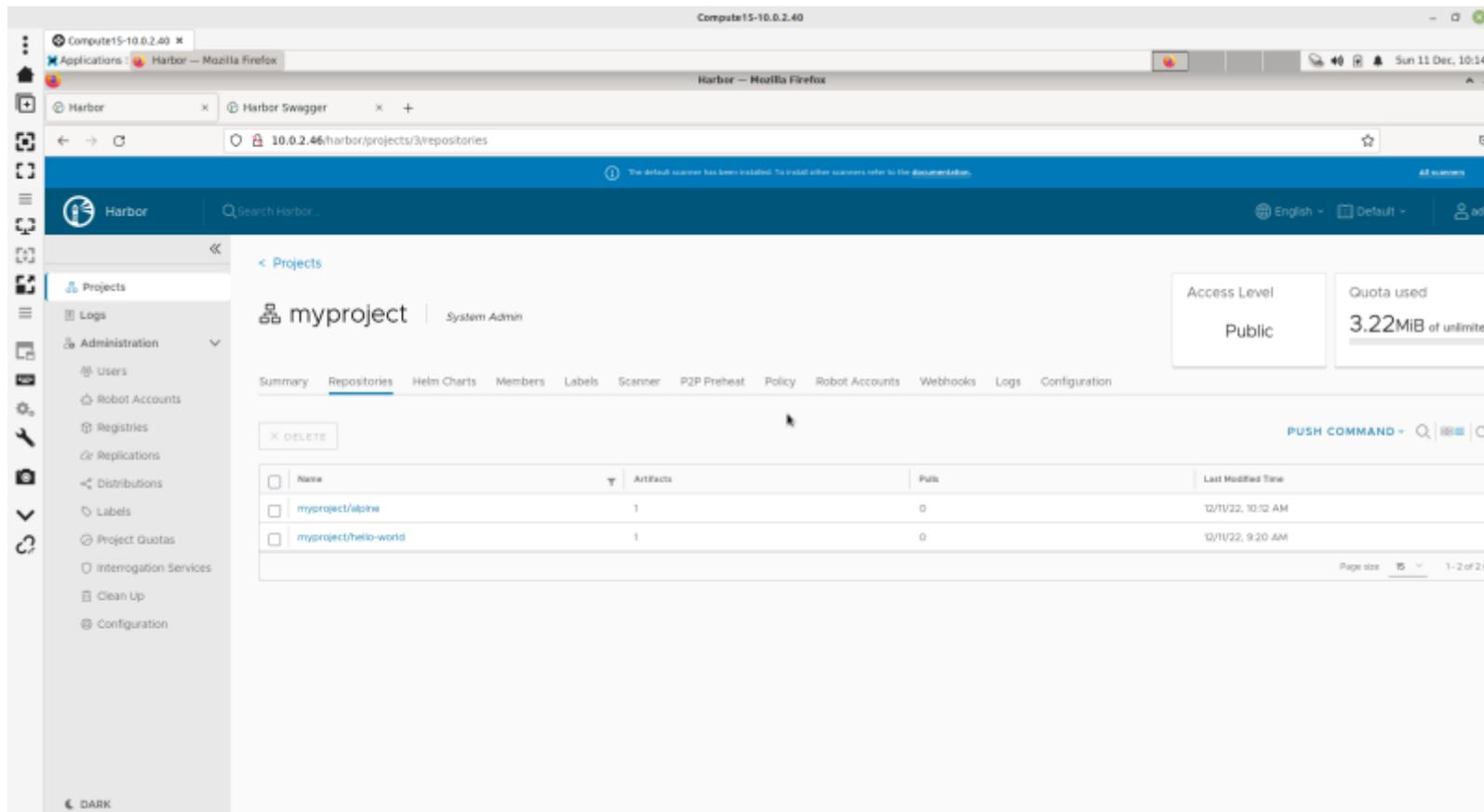
Modifiez le tag de l'image alpine :

```
root@debian11:~/harbor# docker tag alpine 10.0.2.46:80/myproject/alpine
```

Procédez maintenant au **push** :

```
root@debian11:~/harbor# docker push 10.0.2.46:80/myproject/alpine
Using default tag: latest
The push refers to repository [10.0.2.46:80/myproject/alpine]
ded7a220bb05: Pushed
latest: digest: sha256:c0d488a800e4127c334ad20d61d7bc21b4097540327217dfab52262adc02380c size: 528
```

Constatez le résultat dans Harbor :



The screenshot displays the Harbor web interface in a browser window. The address bar shows the URL `10.0.2.46/harbor/projects/3/repositories`. The page title is `myproject` and the user is `System Admin`. The interface includes a sidebar with navigation options like `Projects`, `Logs`, and `Administration`. The main content area shows the `Repositories` tab for `myproject`, with a table listing two repositories: `myproject/alpine` and `myproject/hello-world`. The table columns are `Name`, `Artifacts`, `Pulls`, and `Last Modified Time`. The `myproject/hello-world` repository has 1 artifact and 0 pulls, last modified on 12/11/22 at 9:20 AM. A `PUSH COMMAND` field is visible at the top right of the table.

<input type="checkbox"/>	Name	Artifacts	Pulls	Last Modified Time
<input type="checkbox"/>	myproject/alpine	1	0	12/11/22, 10:12 AM
<input type="checkbox"/>	myproject/hello-world	1	0	12/11/22, 9:20 AM

5.6 - Travailler avec des Artefacts

Cliquez sur **Projects > myproject > myproject/hello-world** puis sur le nom de l'Artefact :

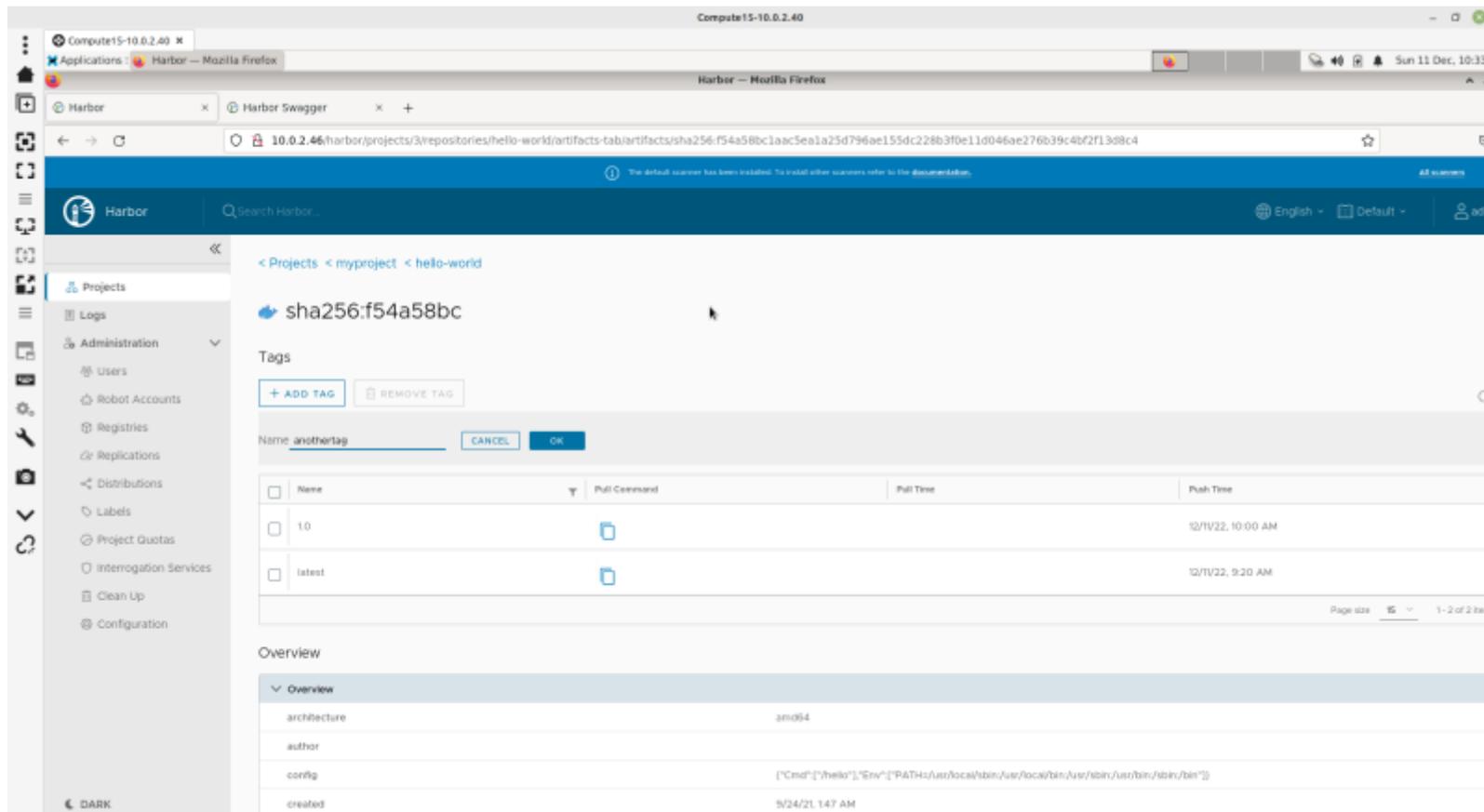
The screenshot displays the Harbor web interface in a browser window. The address bar shows the URL: `10.0.2.46/harbor/projects/3/repositories/hello-world/artifacts-tab/artifacts/sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4`. The page title is "Harbor" and the breadcrumb navigation is "< Projects < myproject < hello-world". The main content area shows the SHA256 hash `sha256:f54a58bc` and a "Tags" section with two buttons: "+ ADD TAG" and "REMOVE TAG". Below the buttons is a table with the following data:

<input type="checkbox"/>	Name	Pull Command	Push Time
<input type="checkbox"/>	1.0	<code>docker pull hello-world:1.0</code>	12/11/22, 10:00 AM
<input type="checkbox"/>	latest	<code>docker pull hello-world:latest</code>	12/11/22, 9:20 AM

Below the table is an "Overview" section with a table of metadata:

Overview	
architecture	amd64
author	
config	<code>["Cmd":["/hello"], "Env":["PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"]]</code>
created	9/24/21, 1:47 AM
os	linux

Cliquez ensuite sur le bouton **ADD TAG** et ajoutez le tag **anothertag** :

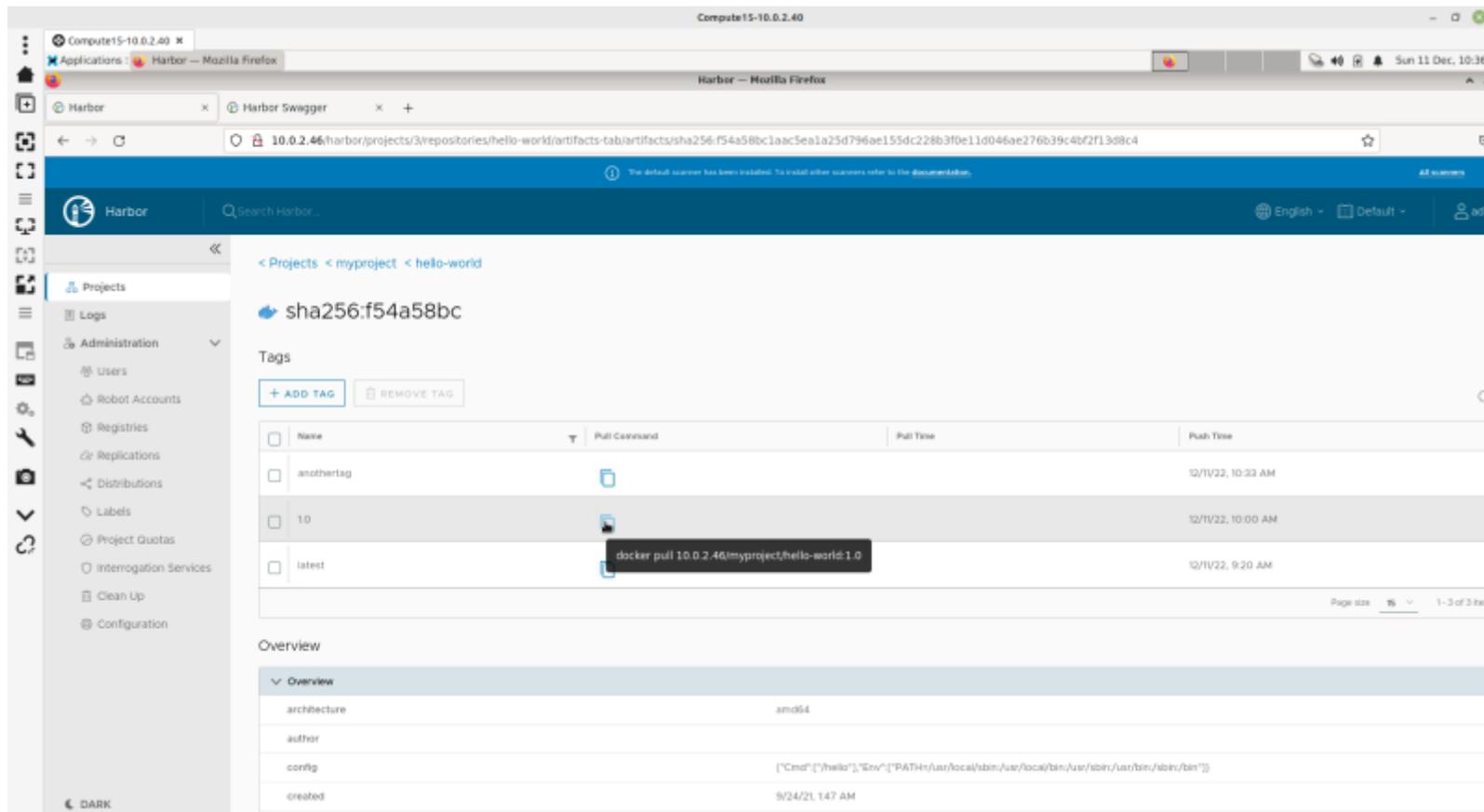


The screenshot shows the Harbor web interface. The breadcrumb navigation is < Projects < myproject < hello-world. The main content area displays the tag sha256:f54a58bc. Below this, there is a 'Tags' section with a table of tags. The table has columns for Name, Pull Command, Pull Time, and Push Time. The '1.0' tag is selected, and the 'Pull Command' column contains a copy icon. Below the table is an 'Overview' section with a table of metadata.

Name	Pull Command	Pull Time	Push Time
1.0			12/11/22, 10:00 AM
latest			12/11/22, 9:20 AM

Overview	
architecture	amd64
author	
config	["Cmd":["hello"],"Env":["PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/bin:/sbin"]]
created	5/24/21, 1:47 AM

Sur la ligne 1.0 et dans la colonne **Pull Command**, cliquez sur l'icône :



The screenshot shows the Harbor web interface. The breadcrumb navigation is < Projects < myproject < hello-world. The main content area shows the tag 'sha256:f54a58bc'. Below this, there is a 'Tags' section with buttons for '+ ADD TAG' and 'REMOVE TAG'. A table lists the tags:

Name	Pull Command	Push Time
anothertag		12/11/22, 10:22 AM
1.0		12/11/22, 10:00 AM
latest		12/11/22, 9:20 AM

A tooltip for the 'latest' tag shows the command: `docker pull 10.0.2.46/myproject/hello-world:1.0`. Below the table is an 'Overview' section with a table of metadata:

Key	Value
architecture	amd64
author	
config	["Cmd":["hello"], "Env":["PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"]]
created	5/24/21 1:47 AM

Revenez à l'interface en ligne de commande et consultez la liste des images :

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
10.0.2.46:80/myproject/alpine	latest	49176f190c7e	2 weeks ago	7.05MB
alpine	latest	49176f190c7e	2 weeks ago	7.05MB
goharbor/harbor-exporter	v2.6.2	ae230e335853	4 weeks ago	95.4MB
goharbor/chartmuseum-photon	v2.6.2	be361eee8a36	4 weeks ago	225MB
goharbor/redis-photon	v2.6.2	84fa3669bb10	4 weeks ago	154MB
goharbor/trivy-adapter-photon	v2.6.2	beef2527c7b2	4 weeks ago	441MB
goharbor/notary-server-photon	v2.6.2	b43e7c963f06	4 weeks ago	112MB
goharbor/notary-signer-photon	v2.6.2	4e9982286316	4 weeks ago	109MB

goharbor/harbor-registryctl	v2.6.2	d403d4e0f990	4 weeks ago	137MB
goharbor/registry-photon	v2.6.2	596e445b18f0	4 weeks ago	77.5MB
goharbor/nginx-photon	v2.6.2	cbbdf28947a5	4 weeks ago	154MB
goharbor/harbor-log	v2.6.2	3ce902023f73	4 weeks ago	161MB
goharbor/harbor-jobservice	v2.6.2	05791a75c8b8	4 weeks ago	250MB
goharbor/harbor-core	v2.6.2	484e2a64616d	4 weeks ago	213MB
goharbor/harbor-portal	v2.6.2	932297030d4e	4 weeks ago	162MB
goharbor/harbor-db	v2.6.2	351b19e366cb	4 weeks ago	225MB
goharbor/prepare	v2.6.2	eecc993c7643	4 weeks ago	164MB
10.0.2.46:80/myproject/hello-world	1.0	feb5d9fea6a5	14 months ago	13.3kB
10.0.2.46:80/myproject/hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB

Supprimez l'image **10.0.2.46:80/myproject/hello-world:1.0** :

```
root@debian11:~/harbor# docker image rm 10.0.2.46:80/myproject/hello-world:1.0
Untagged: 10.0.2.46:80/myproject/hello-world:1.0
```

Consultez la liste des images de nouveau :

```
root@debian11:~/harbor# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
10.0.2.46:80/myproject/alpine	latest	49176f190c7e	2 weeks ago	7.05MB
alpine	latest	49176f190c7e	2 weeks ago	7.05MB
goharbor/harbor-exporter	v2.6.2	ae230e335853	4 weeks ago	95.4MB
goharbor/chartmuseum-photon	v2.6.2	be361eee8a36	4 weeks ago	225MB
goharbor/redis-photon	v2.6.2	84fa3669bb10	4 weeks ago	154MB
goharbor/trivy-adapter-photon	v2.6.2	beef2527c7b2	4 weeks ago	441MB
goharbor/notary-server-photon	v2.6.2	b43e7c963f06	4 weeks ago	112MB
goharbor/notary-signer-photon	v2.6.2	4e9982286316	4 weeks ago	109MB
goharbor/harbor-registryctl	v2.6.2	d403d4e0f990	4 weeks ago	137MB
goharbor/registry-photon	v2.6.2	596e445b18f0	4 weeks ago	77.5MB
goharbor/nginx-photon	v2.6.2	cbbdf28947a5	4 weeks ago	154MB
goharbor/harbor-log	v2.6.2	3ce902023f73	4 weeks ago	161MB

goharbor/harbor-jobservice	v2.6.2	05791a75c8b8	4 weeks ago	250MB
goharbor/harbor-core	v2.6.2	484e2a64616d	4 weeks ago	213MB
goharbor/harbor-portal	v2.6.2	932297030d4e	4 weeks ago	162MB
goharbor/harbor-db	v2.6.2	351b19e366cb	4 weeks ago	225MB
goharbor/prepare	v2.6.2	eecc993c7643	4 weeks ago	164MB
10.0.2.46:80/myproject/hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB

Téléchargez maintenant l'image de **10.0.2.46:80/myproject/hello-world:1.0** à partir de Harbor :

```
root@debian11:~/harbor# docker pull 10.0.2.46:80/myproject/hello-world:1.0
1.0: Pulling from myproject/hello-world
Digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4
Status: Downloaded newer image for 10.0.2.46:80/myproject/hello-world:1.0
10.0.2.46:80/myproject/hello-world:1.0
```

Lister maintenant les images :

```
root@debian11:~/harbor# docker images
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE
10.0.2.46:80/myproject/alpine    latest      49176f190c7e  2 weeks ago  7.05MB
alpine              latest      49176f190c7e  2 weeks ago  7.05MB
goharbor/harbor-exporter        v2.6.2     ae230e335853  4 weeks ago  95.4MB
goharbor/chartmuseum-photon     v2.6.2     be361eee8a36  4 weeks ago  225MB
goharbor/redis-photon           v2.6.2     84fa3669bb10  4 weeks ago  154MB
goharbor/trivy-adapter-photon   v2.6.2     beef2527c7b2  4 weeks ago  441MB
goharbor/notary-server-photon   v2.6.2     b43e7c963f06  4 weeks ago  112MB
goharbor/notary-signer-photon   v2.6.2     4e9982286316  4 weeks ago  109MB
goharbor/harbor-registryctl     v2.6.2     d403d4e0f990  4 weeks ago  137MB
goharbor/registry-photon        v2.6.2     596e445b18f0  4 weeks ago  77.5MB
goharbor/nginx-photon           v2.6.2     cbbdf28947a5  4 weeks ago  154MB
goharbor/harbor-log             v2.6.2     3ce902023f73  4 weeks ago  161MB
goharbor/harbor-jobservice      v2.6.2     05791a75c8b8  4 weeks ago  250MB
goharbor/harbor-core           v2.6.2     484e2a64616d  4 weeks ago  213MB
```

goharbor/harbor-portal	v2.6.2	932297030d4e	4 weeks ago	162MB
goharbor/harbor-db	v2.6.2	351b19e366cb	4 weeks ago	225MB
goharbor/prepare	v2.6.2	eecc993c7643	4 weeks ago	164MB
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB
10.0.2.46:80/myproject/hello-world	1.0	feb5d9fea6a5	14 months ago	13.3kB
10.0.2.46:80/myproject/hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB

Modifiez le tag de l'image téléchargée :

```
root@debian11:~/harbor# docker tag hello-world 10.0.2.46:80/myproject/production:latest
```

```
root@debian11:~/harbor# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
10.0.2.46:80/myproject/alpine	latest	49176f190c7e	2 weeks ago	7.05MB
alpine	latest	49176f190c7e	2 weeks ago	7.05MB
goharbor/harbor-exporter	v2.6.2	ae230e335853	4 weeks ago	95.4MB
goharbor/chartmuseum-photon	v2.6.2	be361eee8a36	4 weeks ago	225MB
goharbor/redis-photon	v2.6.2	84fa3669bb10	4 weeks ago	154MB
goharbor/trivy-adapter-photon	v2.6.2	beef2527c7b2	4 weeks ago	441MB
goharbor/notary-server-photon	v2.6.2	b43e7c963f06	4 weeks ago	112MB
goharbor/notary-signer-photon	v2.6.2	4e9982286316	4 weeks ago	109MB
goharbor/harbor-registryctl	v2.6.2	d403d4e0f990	4 weeks ago	137MB
goharbor/registry-photon	v2.6.2	596e445b18f0	4 weeks ago	77.5MB
goharbor/nginx-photon	v2.6.2	cbbdf28947a5	4 weeks ago	154MB
goharbor/harbor-log	v2.6.2	3ce902023f73	4 weeks ago	161MB
goharbor/harbor-jobservice	v2.6.2	05791a75c8b8	4 weeks ago	250MB
goharbor/harbor-core	v2.6.2	484e2a64616d	4 weeks ago	213MB
goharbor/harbor-portal	v2.6.2	932297030d4e	4 weeks ago	162MB
goharbor/harbor-db	v2.6.2	351b19e366cb	4 weeks ago	225MB
goharbor/prepare	v2.6.2	eecc993c7643	4 weeks ago	164MB
10.0.2.46:80/myproject/hello-world	1.0	feb5d9fea6a5	14 months ago	13.3kB
10.0.2.46:80/myproject/hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB
10.0.2.46:80/myproject/production	latest	feb5d9fea6a5	14 months ago	13.3kB

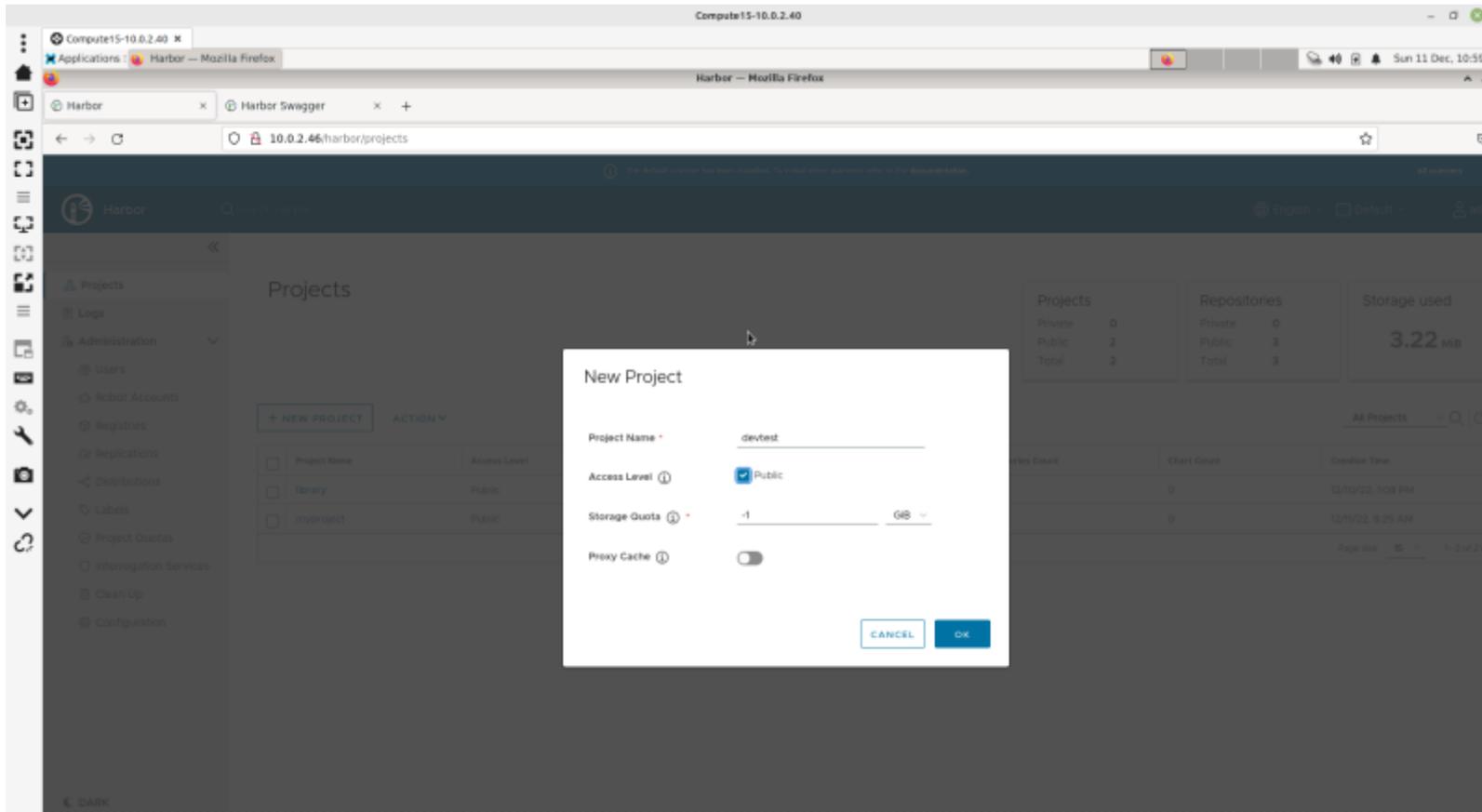
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB
-------------	--------	--------------	---------------	--------

Procédez maintenant au **push** :

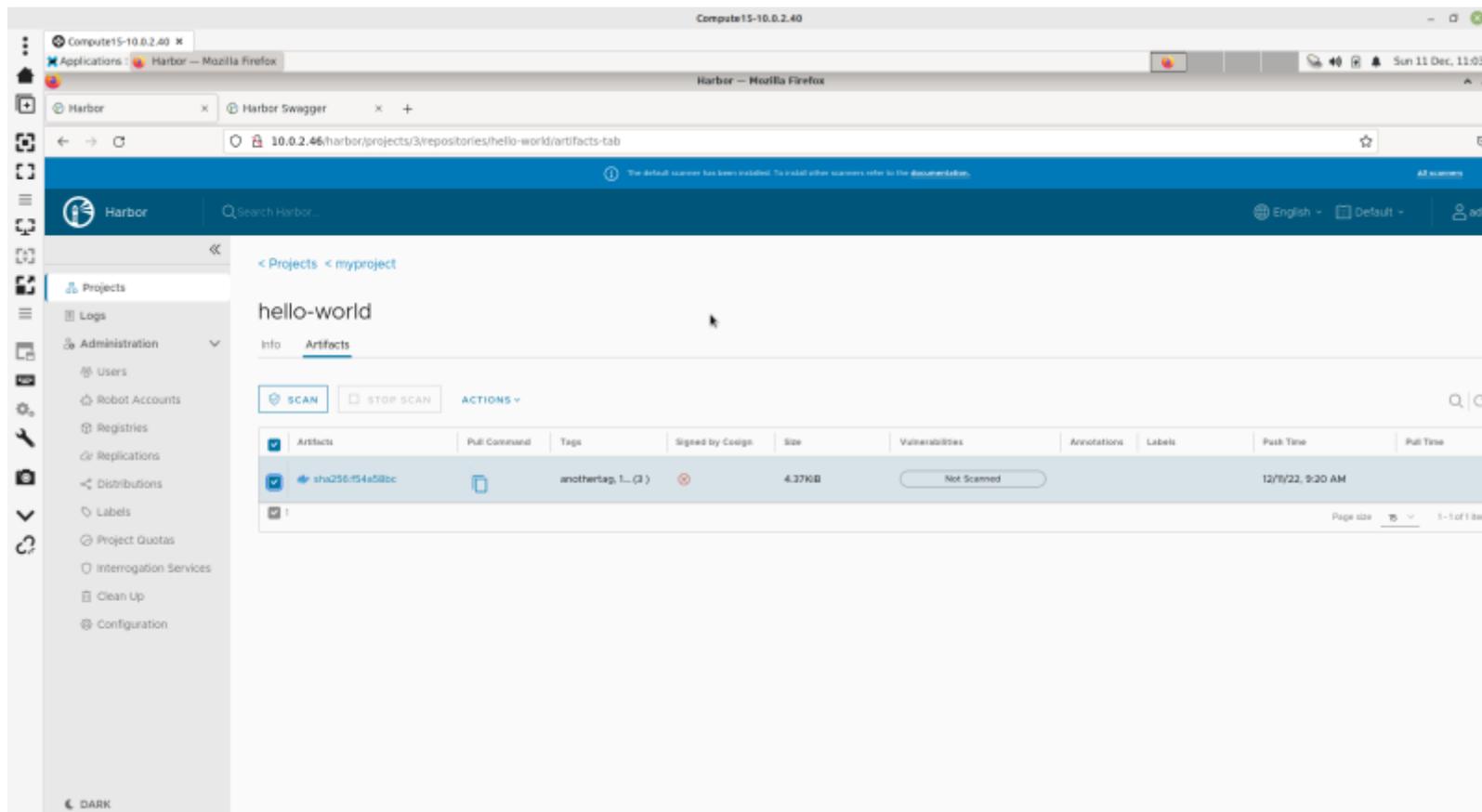
```
root@debian11:~/harbor# docker push 10.0.2.46:80/myproject/production:latest
The push refers to repository [10.0.2.46:80/myproject/production]
e07ee1baac5f: Mounted from myproject/hello-world
latest: digest: sha256:f54a58bclaac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525
```

5.7 - Copier un Artefact d'un Projet vers un autre Projet

Commencez par créer un autre projet appelé **devtest** :



Cliquez sur **Projects > myproject > myproject/hello-world** et cochez l'Artefact :

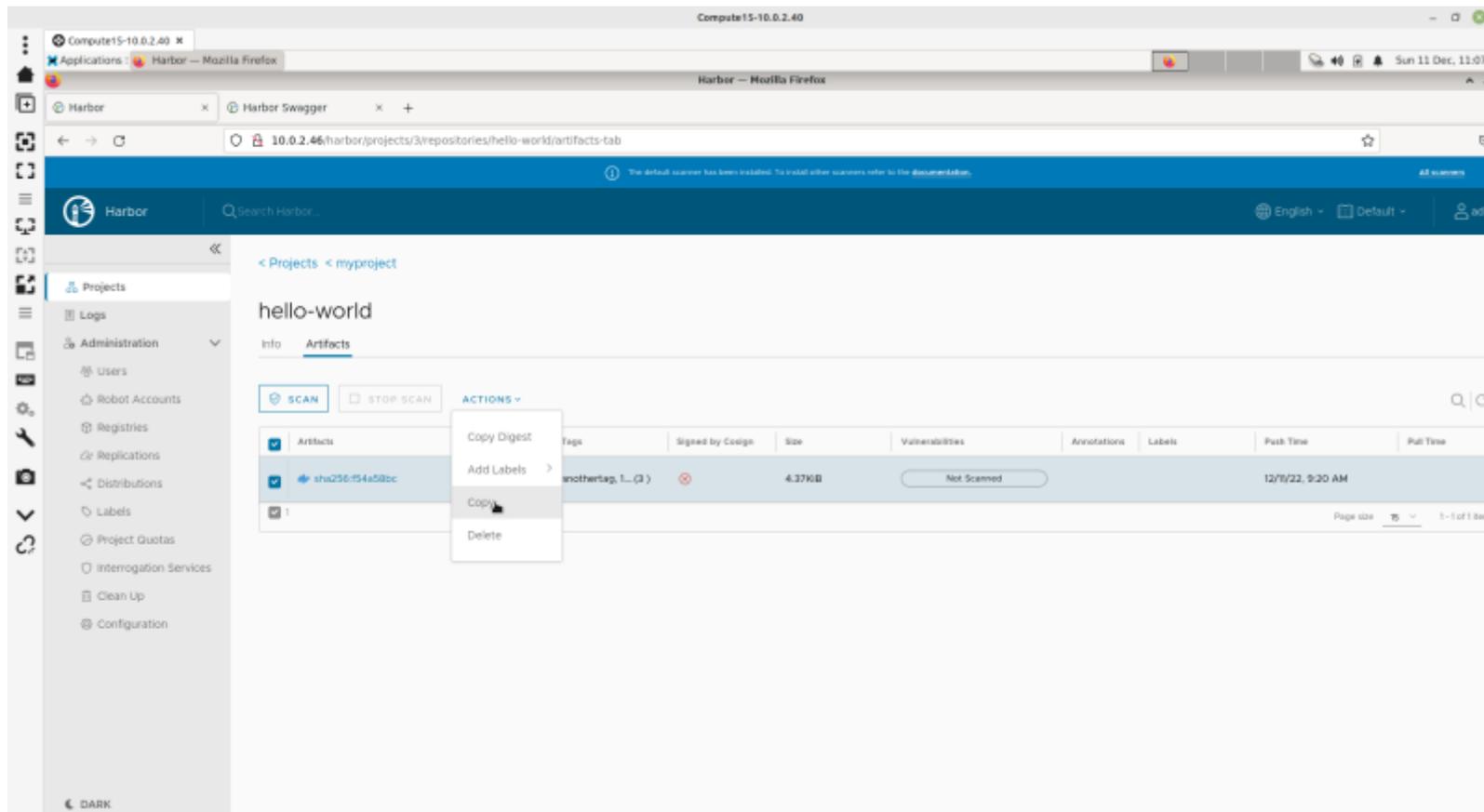


The screenshot shows the Harbor web interface in a browser window. The address bar displays the URL `10.0.2.46/harbor/projects/3/repositories/hello-world/artifacts-tab`. The page title is `hello-world` under the `myproject` project. The left sidebar contains navigation options such as `Projects`, `Logs`, `Administration`, `Users`, `Robot Accounts`, `Registries`, `Replications`, `Distributions`, `Labels`, `Project Quotas`, `Interrogation Services`, `Clean Up`, and `Configuration`. The main content area shows the `Artifacts` tab for the `hello-world` repository. At the top of this section are buttons for `SCAN`, `STOP SCAN`, and `ACTIONS`. Below these is a table of artifacts:

Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
sha256-154a58bc		another tag, 1... (2)		4.37KiB	Not Scanned			12/9/22, 9:20 AM	

At the bottom right of the table, there is a `Page size` dropdown set to `25` and a `1 - 1 of 1 items` indicator.

Cliquez ensuite sur **ACTIONS** > **Copy** :



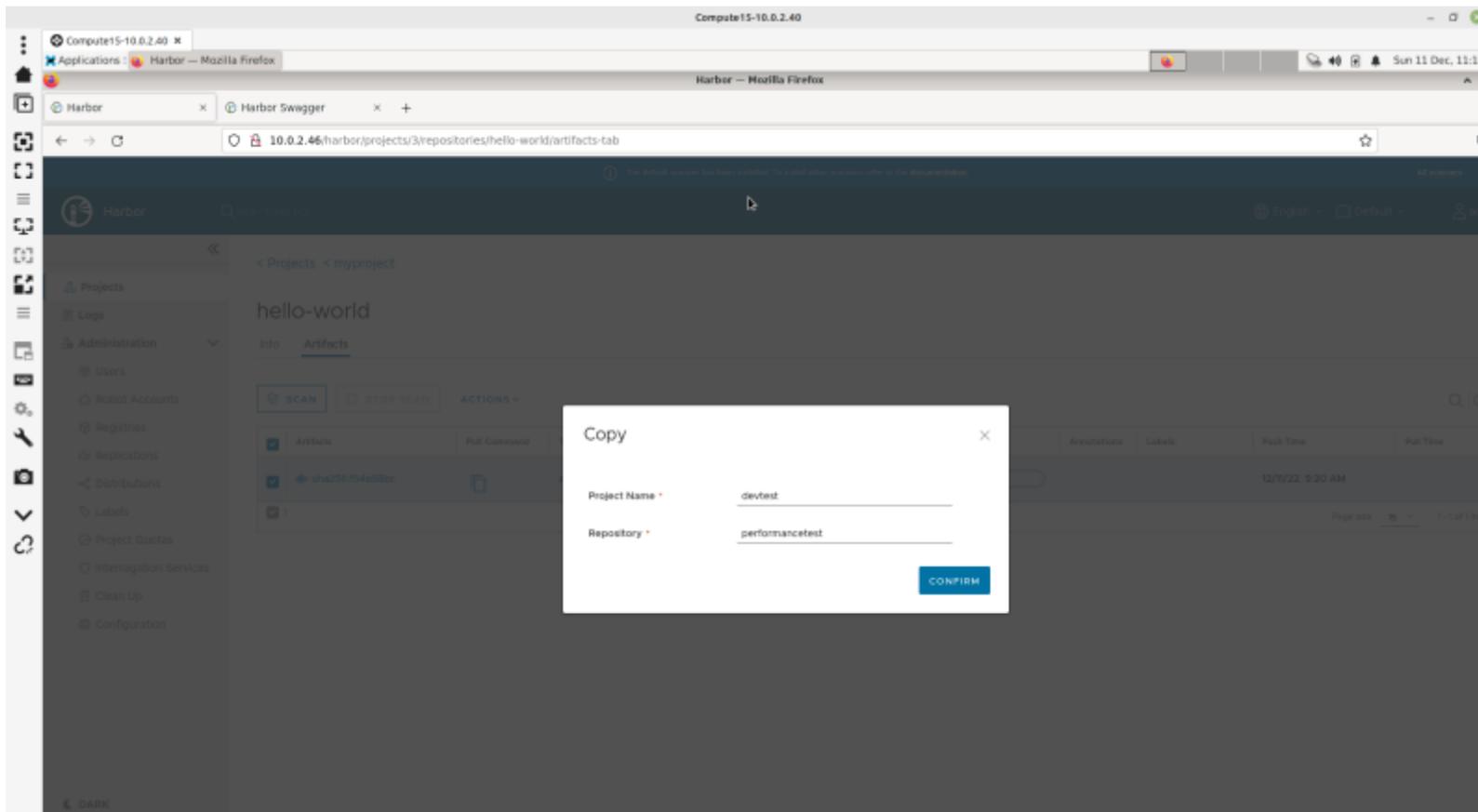
The screenshot shows the Harbor web interface in a browser window. The URL is `10.0.2.46/harbor/projects/3/repositories/hello-world/artifacts-tab`. The page title is "hello-world" and the current view is "Artifacts". A table lists the artifacts:

Artifacts	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
sha256-154a58bc	anothertag, 1. (2)		4.37KiB	Not Scanned			12/9/22, 9:20 AM	

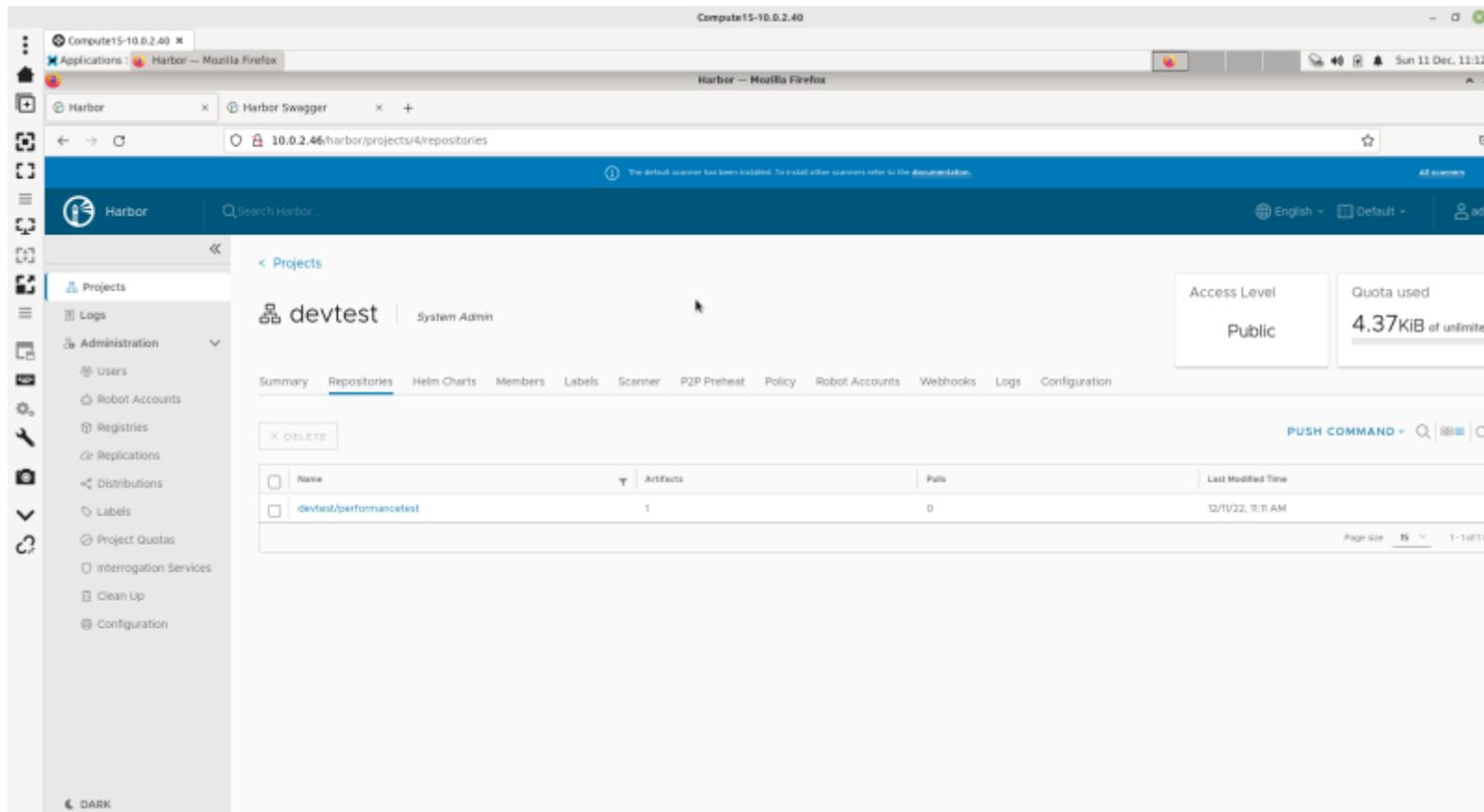
A context menu is open over the first artifact, showing the following options:

- Copy Digest
- Add Labels
- Copy
- Delete

Renseignez les champs et cliquez sur le bouton **CONFIRM** :



Cliquez sur **Projects > devtest** :

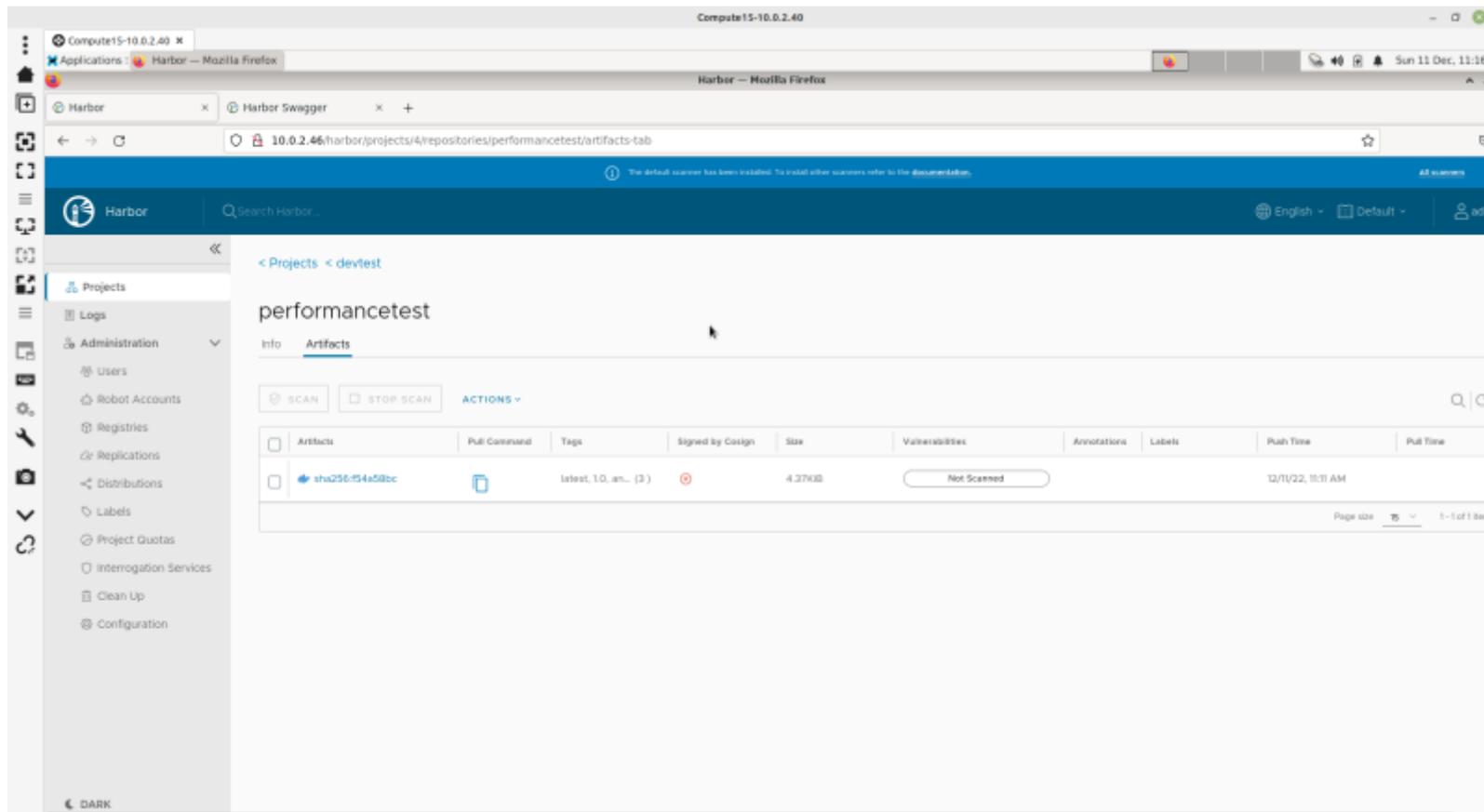


The screenshot shows the Harbor web interface. The browser address bar displays `10.0.2.46/harbor/projects/4/repositories`. The page title is `< Projects`. The project name is `devtest` with the role `System Admin`. The access level is `Public` and the quota used is `4.37 KiB of unlimited`. The page has tabs for `Summary`, `Repositories`, `Helm Charts`, `Members`, `Labels`, `Scanner`, `PIP Preheat`, `Policy`, `Robot Accounts`, `Webhooks`, `Logs`, and `Configuration`. A `DELETE` button is visible. A table lists the repositories:

<input type="checkbox"/>	Name	Artifacts	Pulls	Last Modified Time
<input type="checkbox"/>	devtest/performancetest	1	0	12/11/22, 11:11 AM

Page size: 15, 1-1 of 1

Cliquez ensuite sur **devtest/performancetest** :



The screenshot displays the Harbor web interface in a browser window. The address bar shows the URL `10.0.2.46/harbor/projects/4/repositories/performancetest/artifacts-tab`. The page title is `< Projects < devtest` and the main heading is `performancetest`. The `Artifacts` tab is selected. At the top, there are buttons for `SCAN`, `STOP SCAN`, and `ACTIONS`. Below these is a table of artifacts with the following columns: `Artifacts`, `Pull Command`, `Tags`, `Signed by Cosign`, `Size`, `Vulnerabilities`, `Annotations`, `Labels`, `Push Time`, and `Pull Time`. One artifact is listed with the ID `sha256-154a58bc`, tag `latest, 1.0, an... (3)`, size `4.37KB`, and a `Not Scanned` status. The left sidebar contains navigation options like `Projects`, `Logs`, `Administration`, `Users`, `Robot Accounts`, `Registries`, `Replications`, `Distributions`, `Labels`, `Project Quotas`, `Interrogation Services`, `Clean Up`, and `Configuration`. A `DARK` theme toggle is visible at the bottom left.

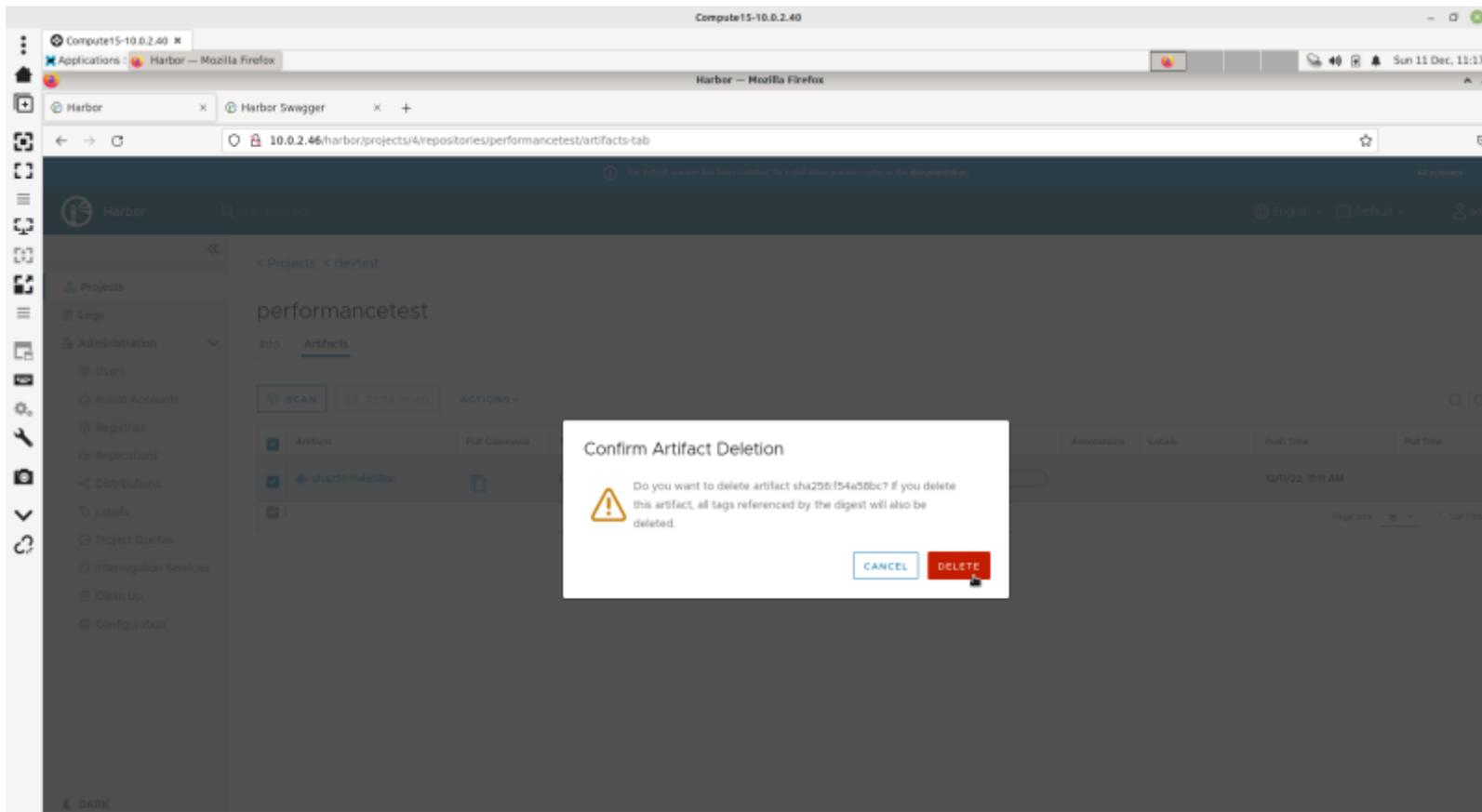
5.8 - Suppression d'un Artefact, d'un Dépôt et d'un Projet

Sélectionnez l'Artefact puis cliquez sur **ACTIONS** > **Delete** :

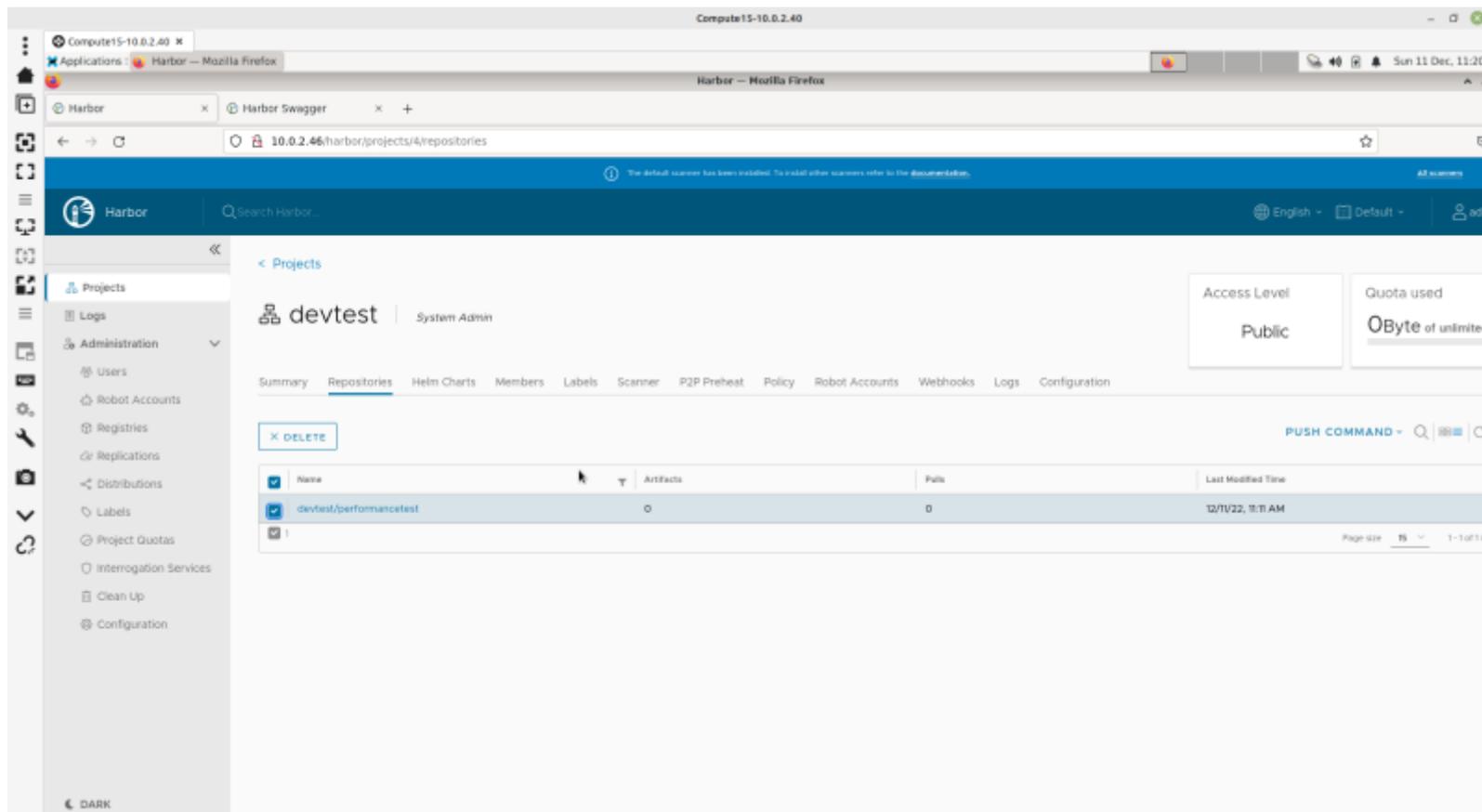
The screenshot shows the Harbor web interface in a browser window. The URL is `10.0.2.46/harbor/projects/4/repositories/performancetest/artifacts-tab`. The page title is "performancetest" and the active tab is "Artifacts". There are buttons for "SCAN" and "STOP SCAN", and an "ACTIONS" dropdown menu. A table lists artifacts with columns for "Tag", "Signed by Cosign", "Size", "Vulnerabilities", "Annotations", "Labels", "Push Time", and "Pull Time". The first artifact is "latest, 1.0, an... (3)" with a size of "4.37KB" and a status of "Not Scanned". A context menu is open over this artifact, showing options: "Copy Digest", "Add Labels", "Copy", and "Delete". The "Delete" option is highlighted.

Tag	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
latest, 1.0, an... (3)		4.37KB	Not Scanned			12/11/22, 11:11 AM	

Confirmez la suppression en cliquant sur le bouton **CONFIRM** :



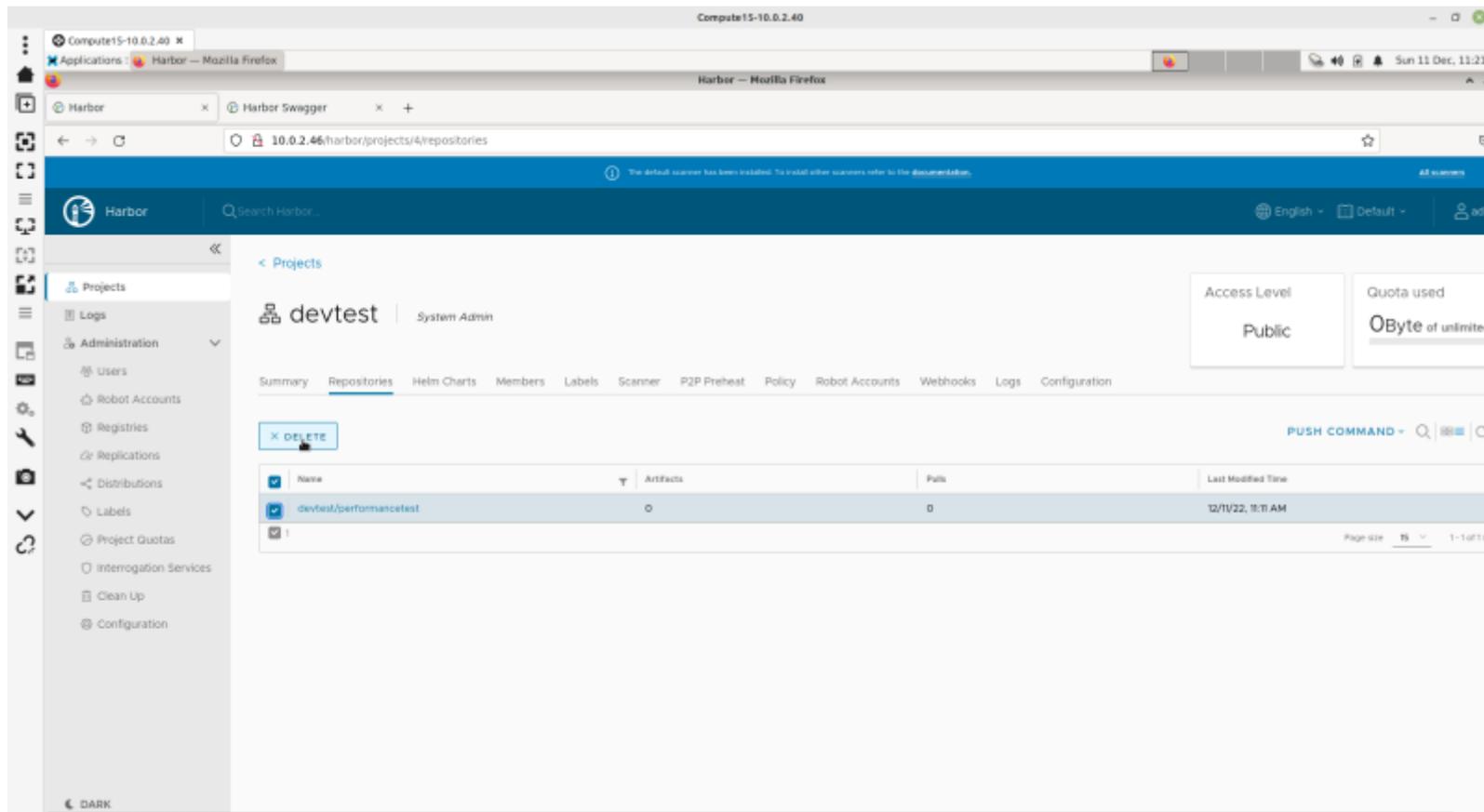
Cliquez sur **devtest** et cochez **devtest/Performancetest** :



The screenshot shows the Harbor web interface for the 'devtest' project. The 'Repositories' tab is selected, showing a table with one repository: 'devtest/performancest'. The table has columns for Name, Artifacts, Pubs, and Last Modified Time. A 'DELETE' button is visible above the table. The interface also shows the project name 'devtest', the user 'System Admin', and various navigation options like Summary, Helm Charts, Members, Labels, Scanner, PIP Preheat, Policy, Robot Accounts, Webhooks, Logs, and Configuration. The 'Access Level' is 'Public' and the 'Quota used' is '0Byte of unlimited'.

Name	Artifacts	Pubs	Last Modified Time
devtest/performancest	0	0	12/11/22, 11:11 AM

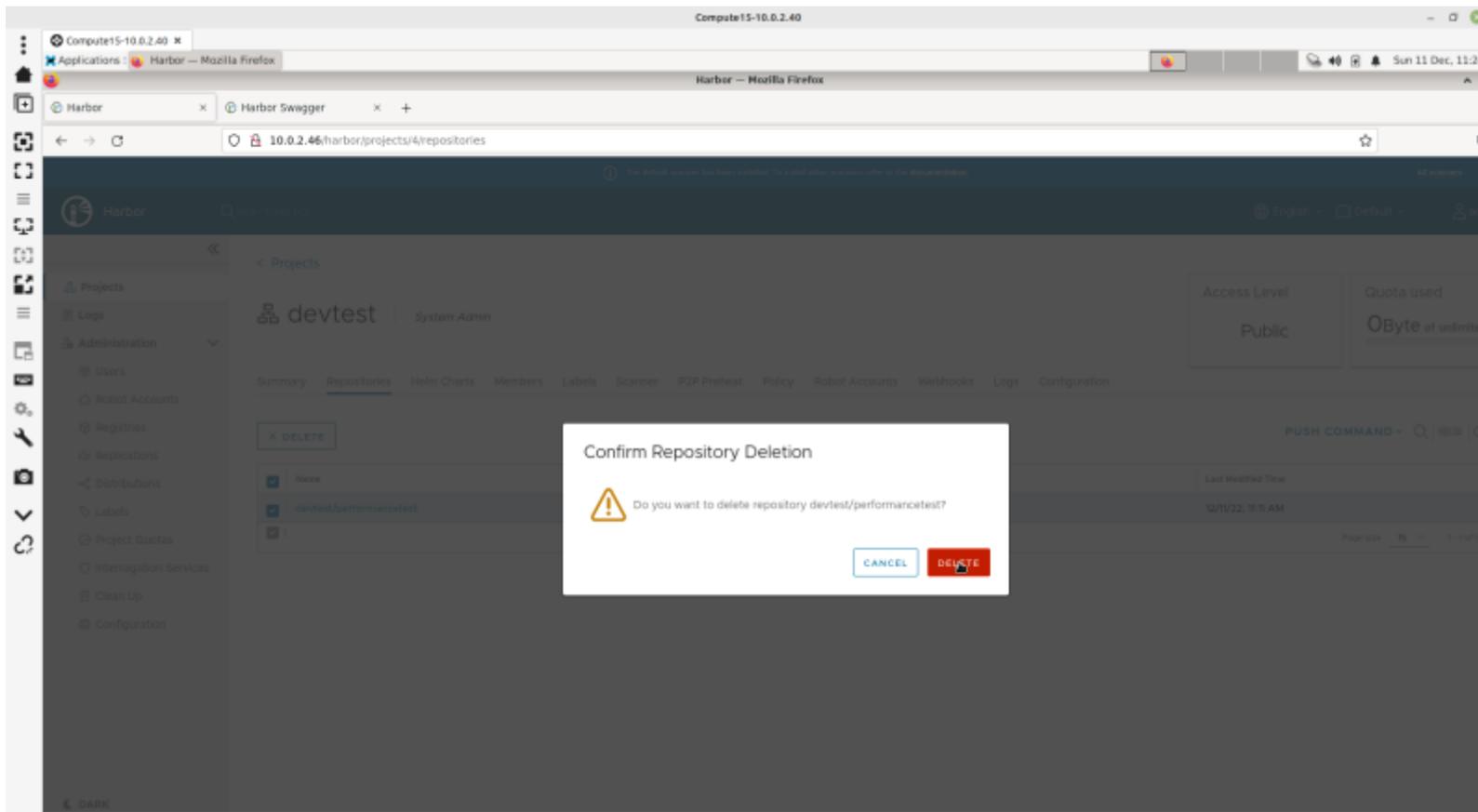
Cliquez sur le bouton **DELETE** :



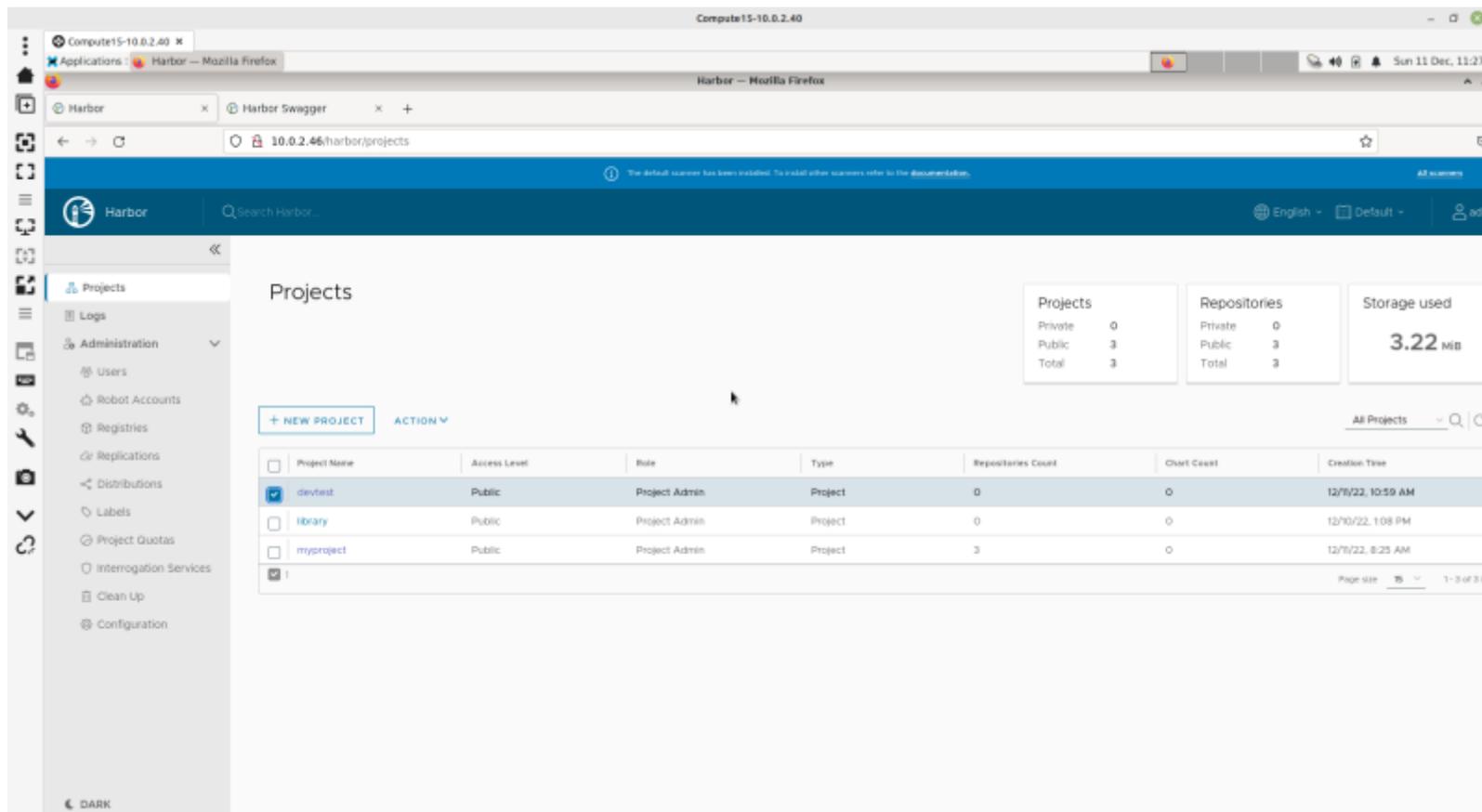
The screenshot shows the Harbor web interface for the 'devtest' project. The 'Repositories' tab is selected, showing a table with one repository: 'devtest/performancest'. The table has columns for Name, Artifacts, Pubs, and Last Modified Time. A 'DELETE' button is visible above the table. The interface also shows the project name 'devtest', the user 'System Admin', and various navigation options like Summary, Helm Charts, Members, Labels, Scanner, PIP Preheat, Policy, Robot Accounts, Webhooks, Logs, and Configuration. The 'Access Level' is 'Public' and the 'Quota used' is '0Byte of unlimited'.

Name	Artifacts	Pubs	Last Modified Time
devtest/performancest	0	0	12/11/22, 11:11 AM

Confirmez la suppression en cliquant sur le bouton **CONFIRM** :



Cliquez sur **Projects** puis cochez **devtest** :



The screenshot shows the Harbor web interface. The main content area is titled "Projects" and features a table of projects. The table has columns for Project Name, Access Level, Role, Type, Repositories Count, Chart Count, and Creation Time. Three projects are listed: "devtest", "library", and "myproject". The "devtest" project is selected. Above the table, there is a "+ NEW PROJECT" button and an "ACTION" dropdown menu. To the right of the table, there are summary statistics for Projects, Repositories, and Storage used. The sidebar on the left contains navigation options such as Projects, Logs, Administration, Users, Robot Accounts, Registries, Replications, Distributions, Labels, Project Quotas, Interrogation Services, Clean Up, and Configuration. The top navigation bar includes a search bar, language settings, and a user profile icon.

Project Name	Access Level	Role	Type	Repositories Count	Chart Count	Creation Time
<input checked="" type="checkbox"/> devtest	Public	Project Admin	Project	0	0	12/9/22, 10:59 AM
<input type="checkbox"/> library	Public	Project Admin	Project	0	0	12/9/22, 1:08 PM
<input type="checkbox"/> myproject	Public	Project Admin	Project	3	0	12/9/22, 8:25 AM

Summary Statistics:

Category	Private	Public	Total
Projects	0	3	3
Repositories	0	3	3
Storage used			3.22 MiB

Cliquez sur **ACTION** puis sur **Delete** :

The screenshot displays the Harbor web interface. The main content area is titled "Projects". On the right, there are three summary cards:

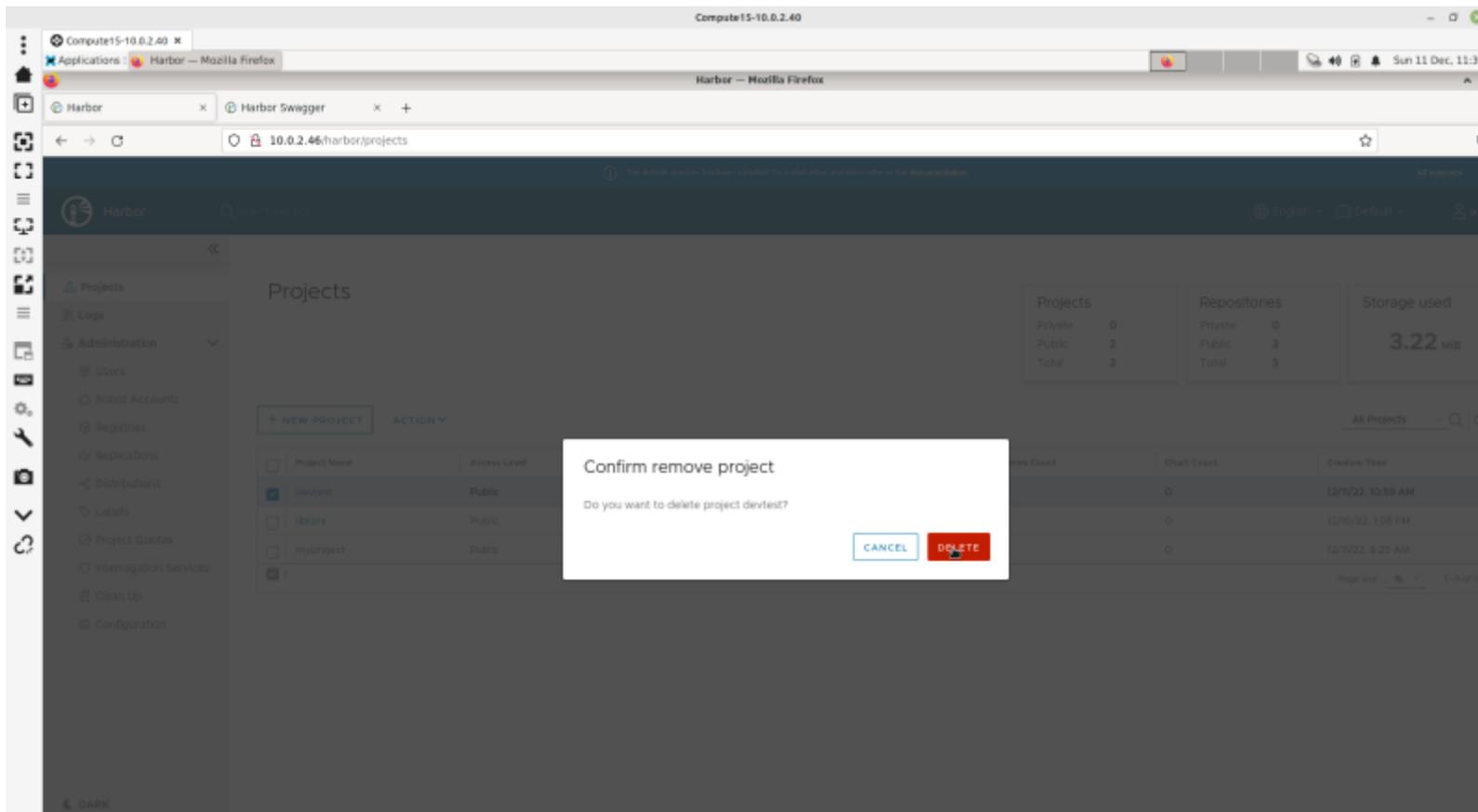
- Projects:** Private: 0, Public: 3, Total: 3
- Repositories:** Private: 0, Public: 3, Total: 3
- Storage used:** 3.22 MiB

A table lists the projects. The "devtest" project is selected, and a context menu is open over it, showing "Export CVEs" and "Delete" options.

<input type="checkbox"/>	Project Name	Level	Role	Type	Repositories Count	Chart Count	Creation Time
<input checked="" type="checkbox"/>	devtest		Project Admin	Project	0	0	12/9/22, 10:59 AM
<input type="checkbox"/>	library		Project Admin	Project	0	0	12/9/22, 1:08 PM
<input type="checkbox"/>	myproject	Public	Project Admin	Project	3	0	12/9/22, 8:25 AM

At the bottom right of the table, it says "Page size 15 1 - 3 of 3 Items".

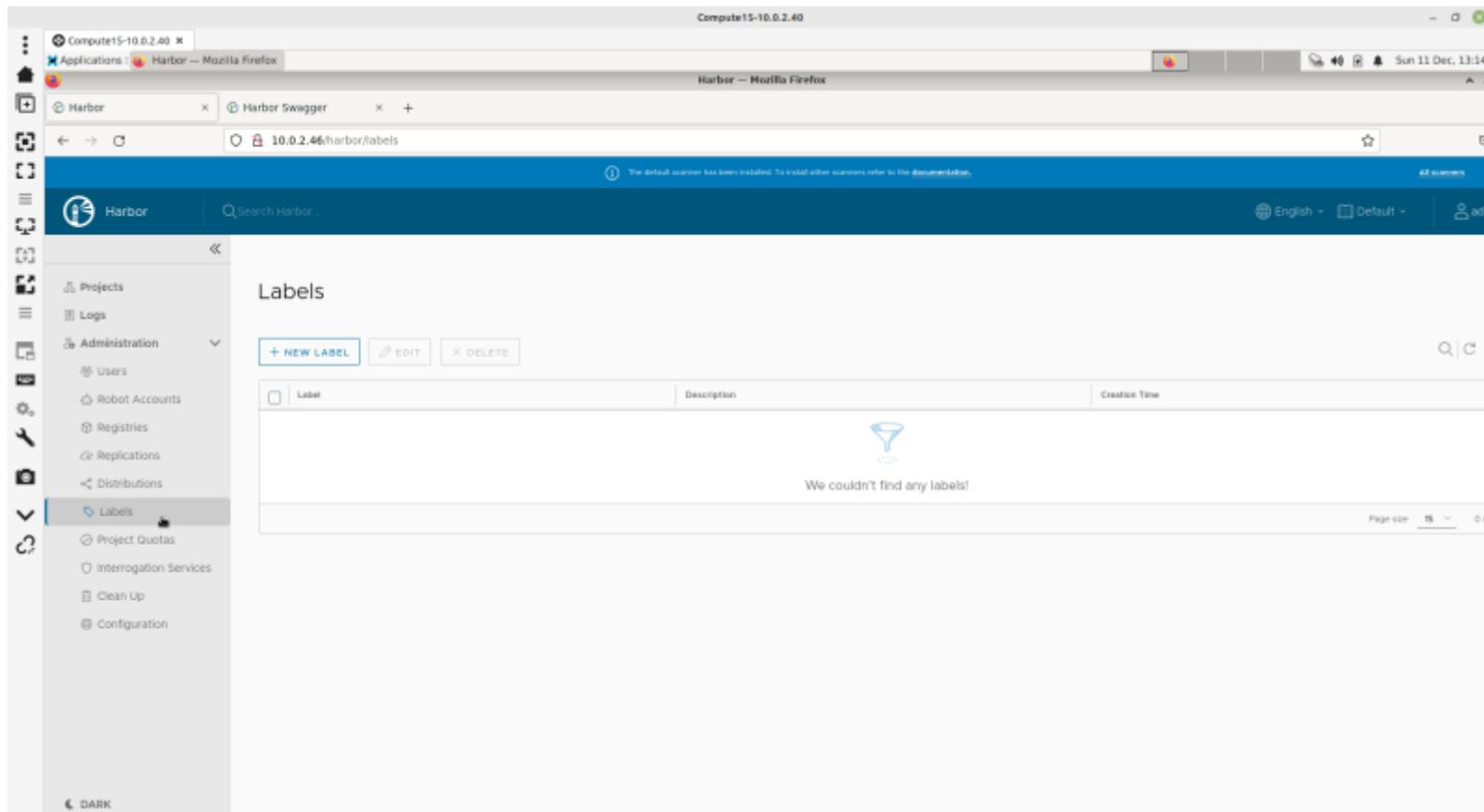
Confirmez la suppression en cliquant sur le bouton **CONFIRM** :



5.8 - Utilisation de Labels

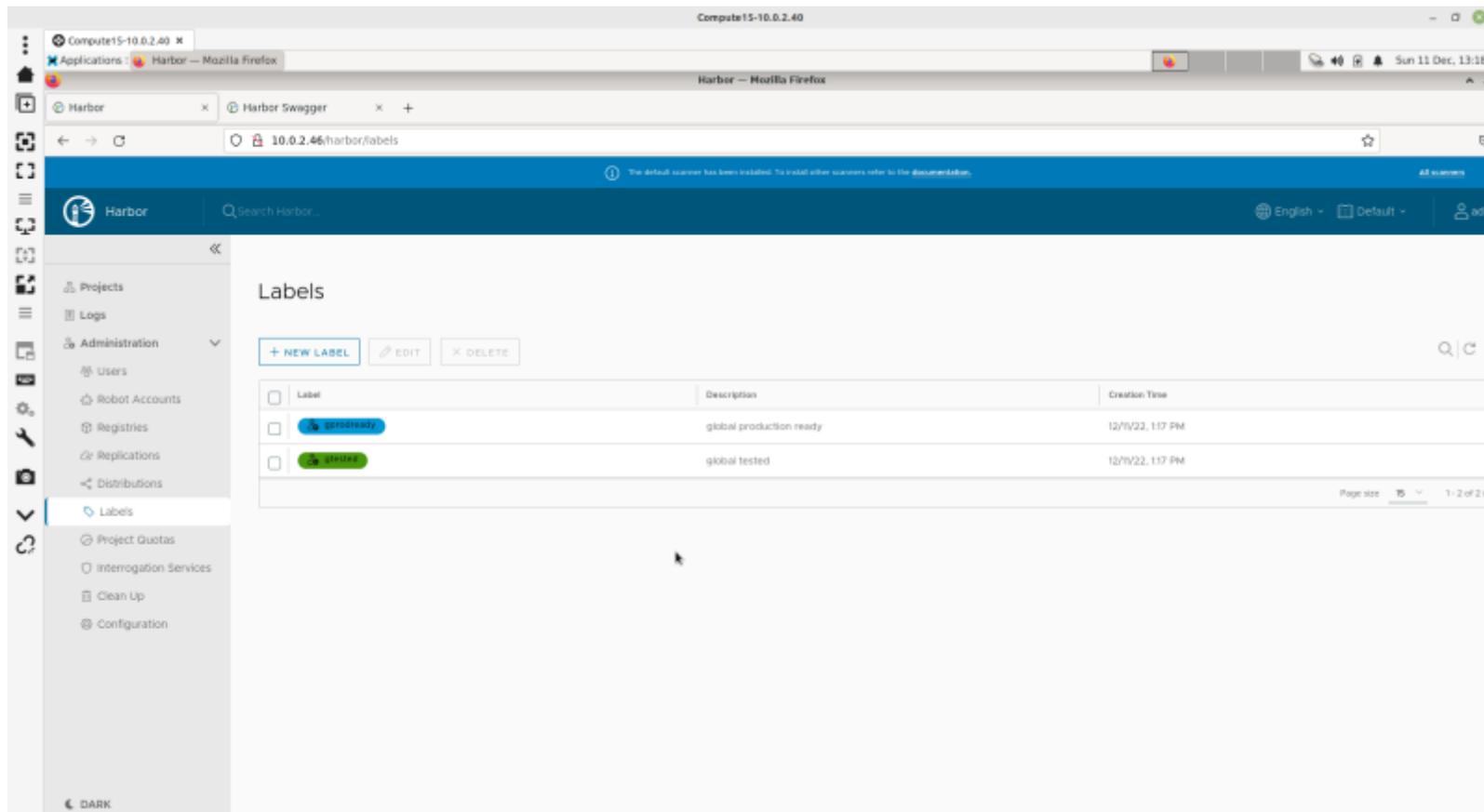
Harbor gère des **Labels** ou étiquettes. Ces étiquettes peuvent être globales ou spécifiques à un Projet. Quand une étiquette est spécifique, celle-ci est utilisée pour marquer les Artefacts pour une utilisation spécifique. Les étiquettes globales ne peuvent être gérées que par l'Administrateur de Harbor.

Cliquez sur **Labels** dans Harbor :



Cliquez sur **NEW LABEL** et créez les deux étiquettes suivantes :

- **gtested** - global tested
- **gprodrady** global production ready

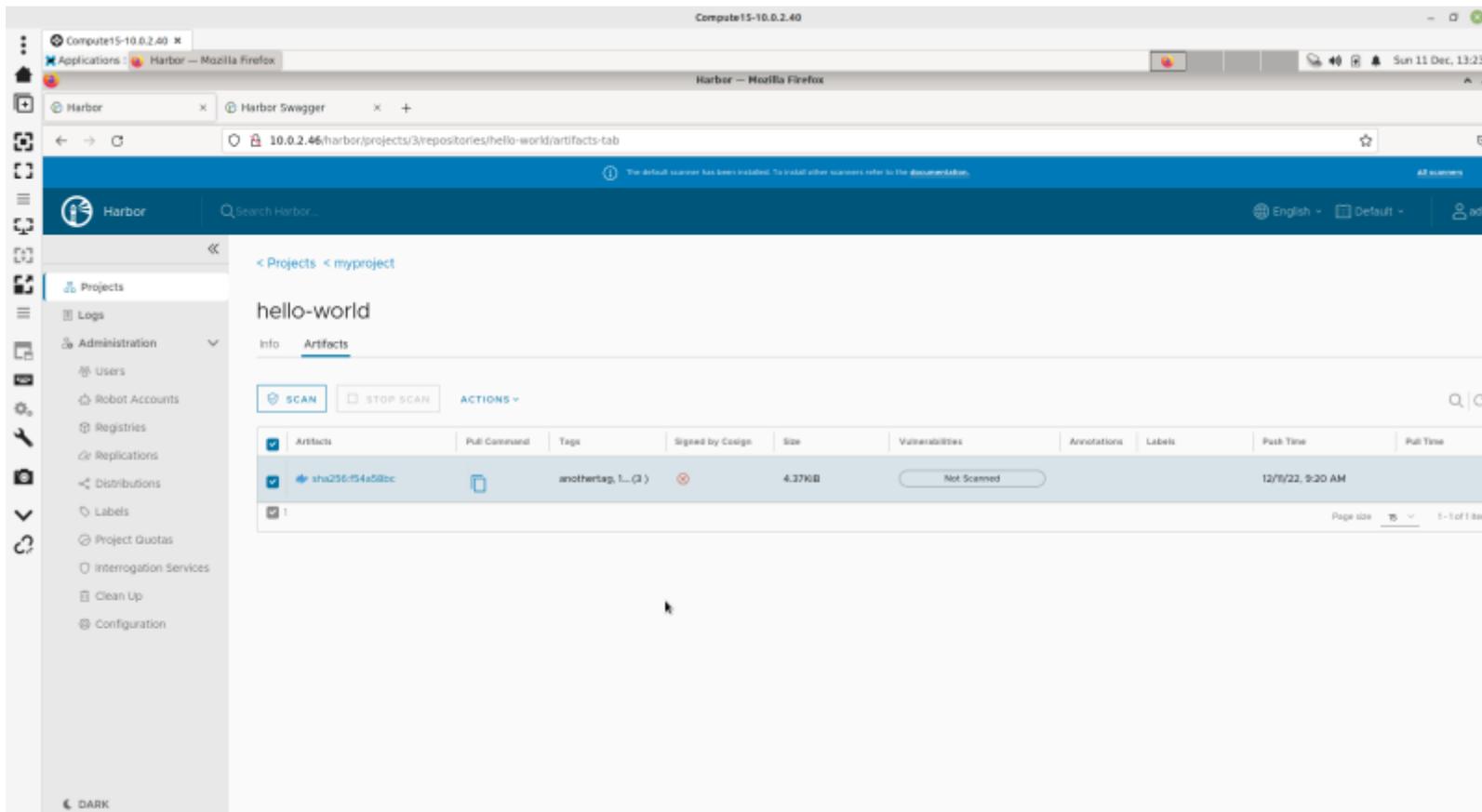


The screenshot shows the Harbor web interface in a browser window. The address bar displays the URL `10.0.2.46/harbor/labels`. The interface includes a sidebar menu on the left with the following items: Projects, Logs, Administration (Users, Robot Accounts, Registries, Replications, Distributions, Labels, Project Quotas, Interrogation Services, Clean Up, Configuration), and a DARK theme toggle at the bottom. The main content area is titled "Labels" and features three buttons: "+ NEW LABEL", "EDIT", and "X DELETE". Below these buttons is a table with the following data:

<input type="checkbox"/>	Label	Description	Creation Time
<input type="checkbox"/>	production	global production ready	12/11/22, 1:17 PM
<input type="checkbox"/>	staging	global tested	12/11/22, 1:17 PM

At the bottom right of the table, there is a "Page size" dropdown set to 25 and a "1 - 2 of 2 items" indicator.

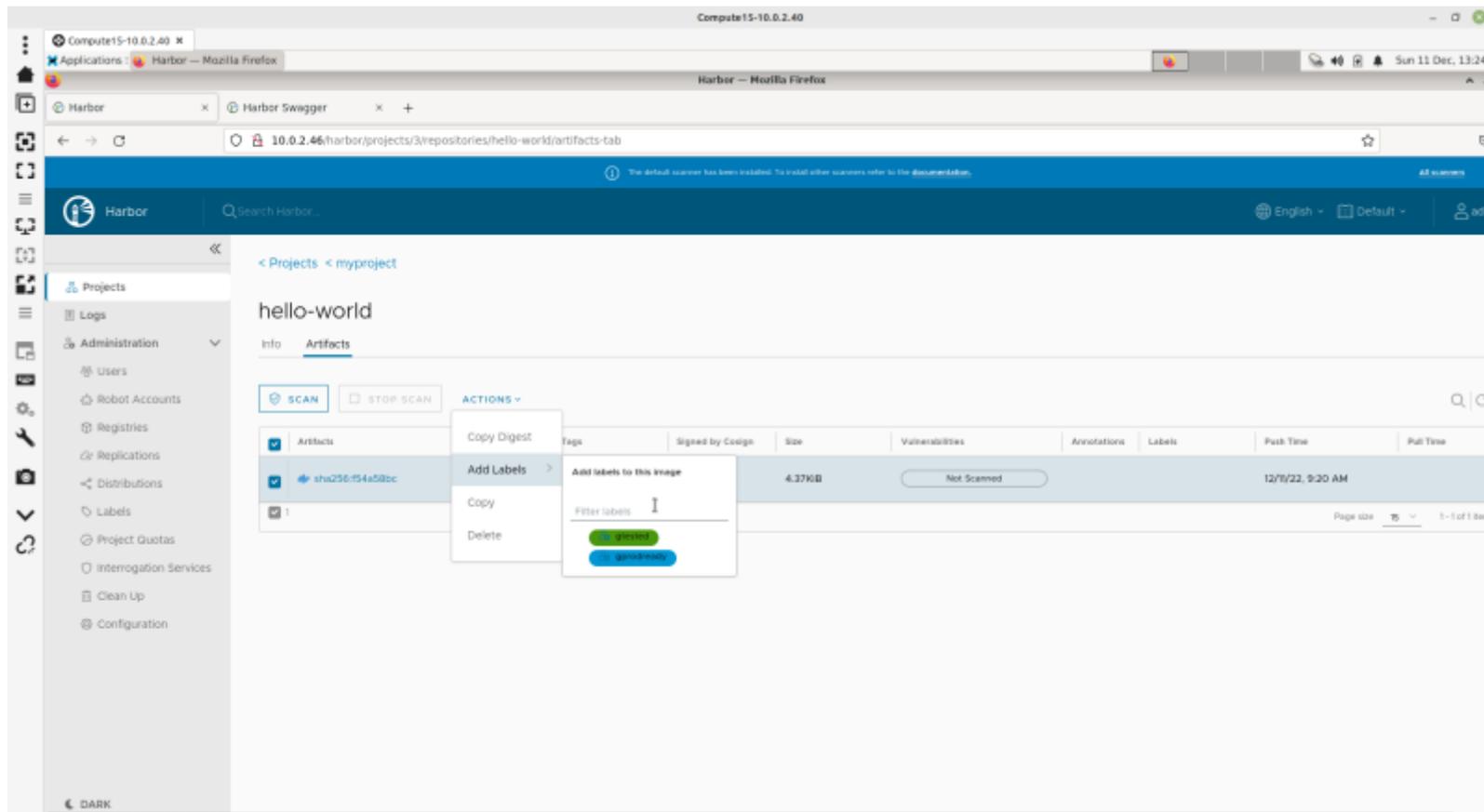
Cliquez sur **Projects > myproject > myproject/hello-world** puis sélectionnez l'Artefact :



The screenshot shows the Harbor web interface in a browser window. The URL is `10.0.2.46/harbor/projects/3/repositories/hello-world/artifacts-tab`. The page title is "hello-world" and the active tab is "Artifacts". A table lists artifacts with columns: Artifacts, Pull Command, Tags, Signed by Cosign, Size, Vulnerabilities, Annotations, Labels, Push Time, and Pull Time. One artifact is listed with ID `sha256-154a58bc`, tag `anothertag, 1. (2)`, size `4.37KiB`, and push time `12/9/22, 9:20 AM`. The "Vulnerabilities" column shows "Not Scanned".

Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
sha256-154a58bc		anothertag, 1. (2)		4.37KiB	Not Scanned			12/9/22, 9:20 AM	

Cliquez sur **ACTIONS** > **Add Labels** puis sélectionnez **gtested** :



The screenshot shows the Harbor web interface in a browser window. The page is titled 'hello-world' and is under the 'Artifacts' tab. A table lists artifacts, with one artifact selected: 'sha256-154a58bc'. The 'ACTIONS' menu is open, showing options like 'Copy Digest', 'Add Labels', 'Copy', and 'Delete'. The 'Add Labels' option is highlighted, and a sub-menu is open with the text 'Add labels to this image' and a 'Filter labels' input field. Two buttons are visible: 'github' (green) and 'github:prod' (blue). The left sidebar shows navigation options like 'Projects', 'Logs', 'Administration', 'Users', 'Robot Accounts', 'Registries', 'Replications', 'Distributions', 'Labels', 'Project Quotas', 'Interrogation Services', 'Clean Up', and 'Configuration'. The top navigation bar includes 'Harbor', a search bar, and language settings.

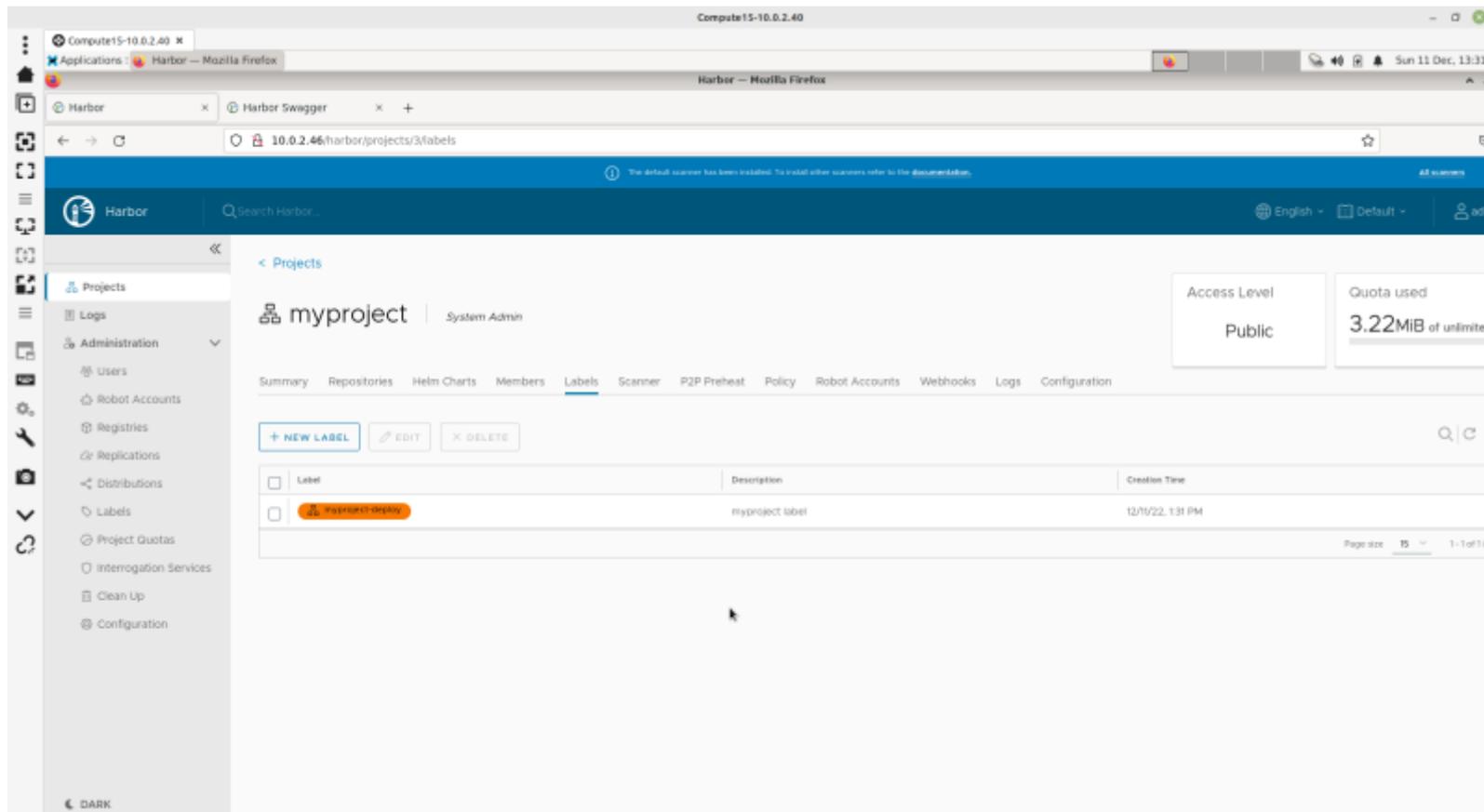
Cliquez maintenant sur **myproject > Labels** :

The screenshot shows the Harbor web interface in a browser window. The URL is `10.0.2.46/harbor/projects/3/repositories`. The page displays the 'Labels' tab for the 'myproject' project, which is managed by 'System Admin'. The 'Access Level' is 'Public' and the 'Quota used' is '3.22MiB of unlimited'. A table lists the existing labels:

<input type="checkbox"/>	Name	Artifacts	Pulls	Last Modified Time
<input type="checkbox"/>	myproject/production	1	0	12/11/22, 10:55 AM
<input type="checkbox"/>	myproject/alpine	1	0	12/11/22, 10:12 AM
<input type="checkbox"/>	myproject/hello-world	1	0	12/11/22, 9:20 AM

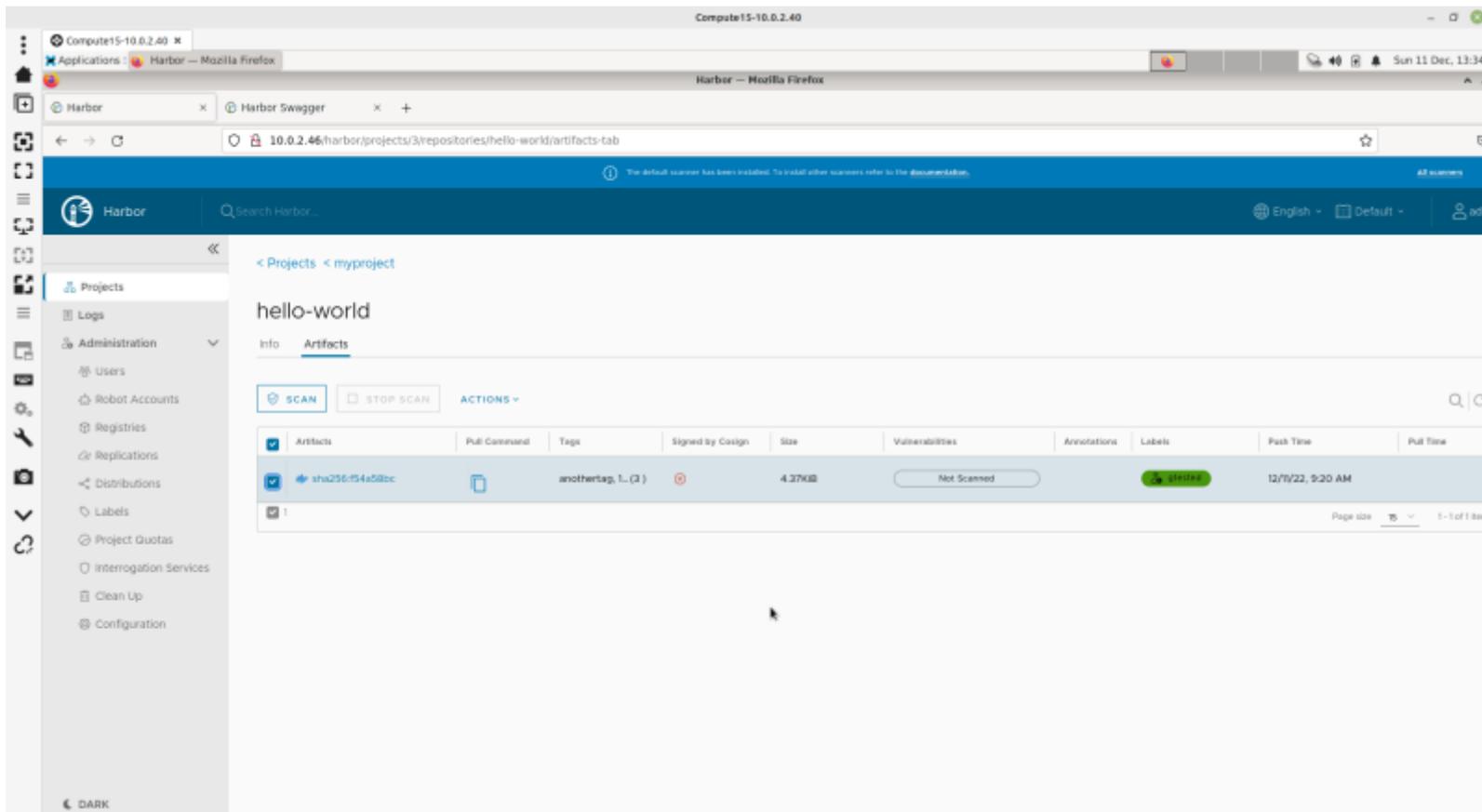
At the bottom right of the table, it shows 'Page size 15' and '1 - 3 of 3 items'.

Cliquez ensuite sur **NEW LABEL** et créez ensuite une étiquette spécifique au projet appelé **myproject-deploy** ayant une description **myproject label** :



The screenshot displays the Harbor web interface in a browser window. The address bar shows the URL `10.0.2.46/harbor/projects/3/labels`. The page title is `myproject | System Admin`. The left sidebar contains a navigation menu with options like `Projects`, `Logs`, `Administration`, `Users`, `Robot Accounts`, `Registries`, `Replications`, `Distributions`, `Labels`, `Project Quotas`, `Interrogation Services`, `Clean Up`, and `Configuration`. The main content area shows the `Labels` tab selected, with a table listing labels. The table has columns for `Label`, `Description`, and `Creation Time`. One label is visible: `myproject:hello-world` with a description of `myproject label` and a creation time of `12/11/22, 1:31 PM`. Above the table are buttons for `+ NEW LABEL`, `EDIT`, and `DELETE`. In the top right corner, there are two summary boxes: `Access Level: Public` and `Quota used: 3.22MiB of unlimited`. The bottom right corner shows `Page size: 15` and `1 - 1 of 1 items`.

Cliquez sur **Repositories > myproject/hello-world** et sélectionnez l'Artefact :



The screenshot shows the Harbor web interface in a browser window. The URL is `10.0.2.46/harbor/projects/3/repositories/hello-world/artifacts-tab`. The page title is "hello-world" and the active tab is "Artifacts". A table lists artifacts with columns: Artifacts, Pull Command, Tags, Signed by Cosign, Size, Vulnerabilities, Annotations, Labels, Push Time, and Pull Time. One artifact is visible with SHA256 `sha256-154a58bc`, tag `anothertag, 1. (2)`, size `4.27KB`, and push time `12/19/22, 9:20 AM`. The "Vulnerabilities" column shows "Not Scanned". The "Labels" column shows a green "myproject" label. The "ACTIONS" dropdown menu is visible above the table.

Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
sha256-154a58bc		anothertag, 1. (2)		4.27KB	Not Scanned		myproject	12/19/22, 9:20 AM	

Cliquez sur **ACTIONS** > **Add Labels** et sélectionnez l'étiquette **myproject-deploy** :

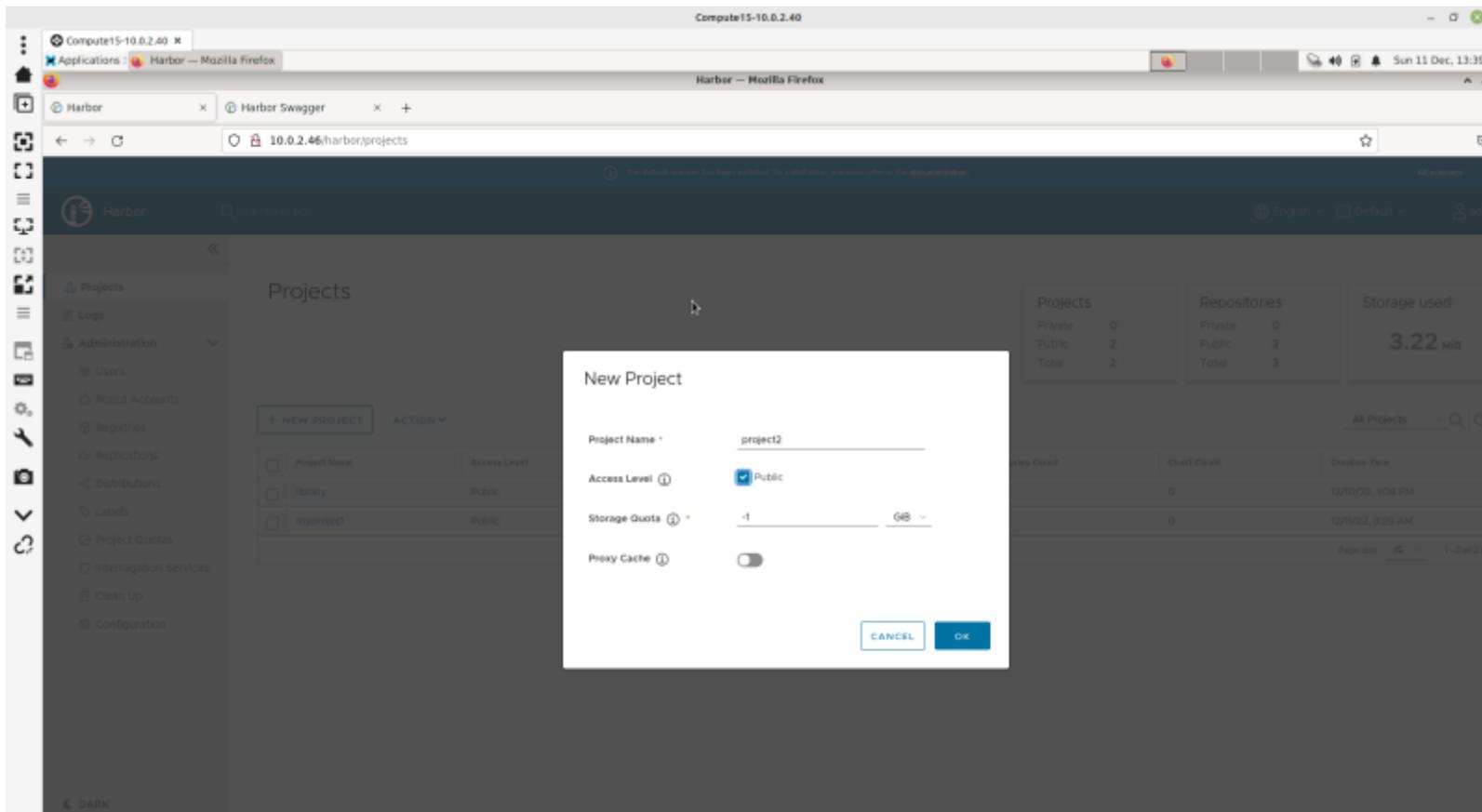
The screenshot shows the Harbor web interface. The main content area displays the 'hello-world' repository page under the 'myproject' project. The 'Artifacts' tab is active, showing a table of artifacts. The table has the following columns: Artifacts, Pull Command, Tags, Signed by Cosign, Size, Vulnerabilities, Annotations, Labels, Push Time, and Pull Time. One artifact is listed with the tag 'another tag, 1. (3)', a size of 4.37KB, and a 'Not Scanned' status. The interface includes a sidebar with navigation options like Projects, Logs, Administration, and a top navigation bar with a search bar and user profile.

Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
<input type="checkbox"/>		another tag, 1. (3)		4.37KB	Not Scanned			12/19/22, 9:20 AM	

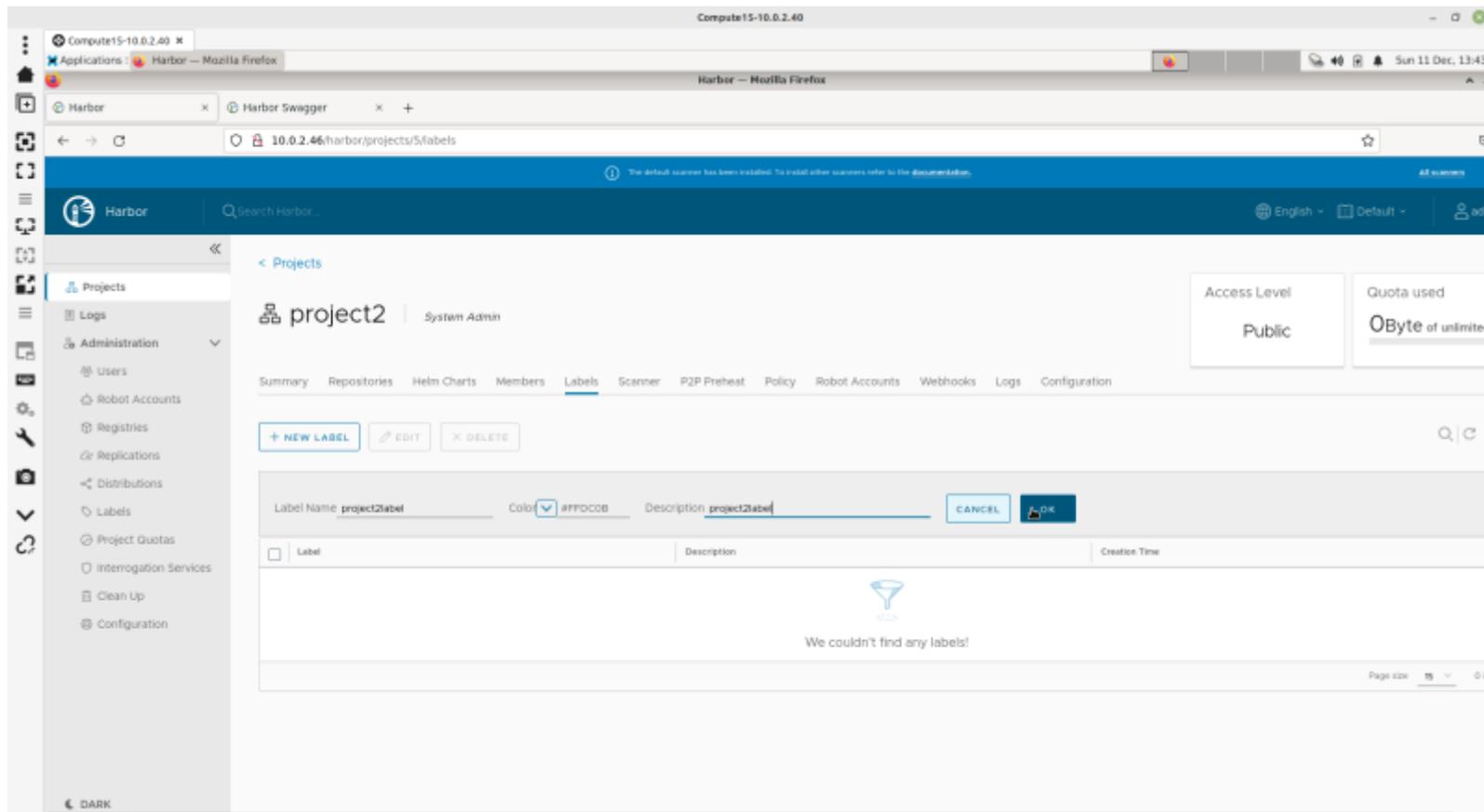
Pour visualiser toutes les étiquettes, cliquez sur les 3 points à droite de **gtested** :

The screenshot displays the Harbor web interface. The browser address bar shows the URL `10.0.2.46/harbor/projects/3/repositories/hello-world/artifacts-tab`. The page title is `hello-world` and the sub-page is `Artifacts`. A sidebar on the left contains navigation links: Projects, Logs, Administration (Users, Robot Accounts, Registries, Replications, Distributions, Labels, Project Quotas, Interrogation Services, Clean Up, Configuration), and a DARK mode toggle. The main content area features a table of artifacts. Above the table are buttons for `SCAN`, `STOP SCAN`, and `ACTIONS`. The table has the following columns: Artifacts, Pull Command, Tags, Signed by Cosign, Size, Vulnerabilities, Annotations, Labels, Is Time, and Pull Time. One artifact is listed with the ID `sha256:f54a59bc`, tag `another-tag, 1. (3)`, size `4.37KB`, and a `Not Scanned` status. A `SCAN` button is visible next to this artifact. A tooltip is visible over the `SCAN` button, showing `SCAN` and `STOP SCAN` options.

Cliquez sur **Projects > NEW PROJECT** et créez le projet **project2** :

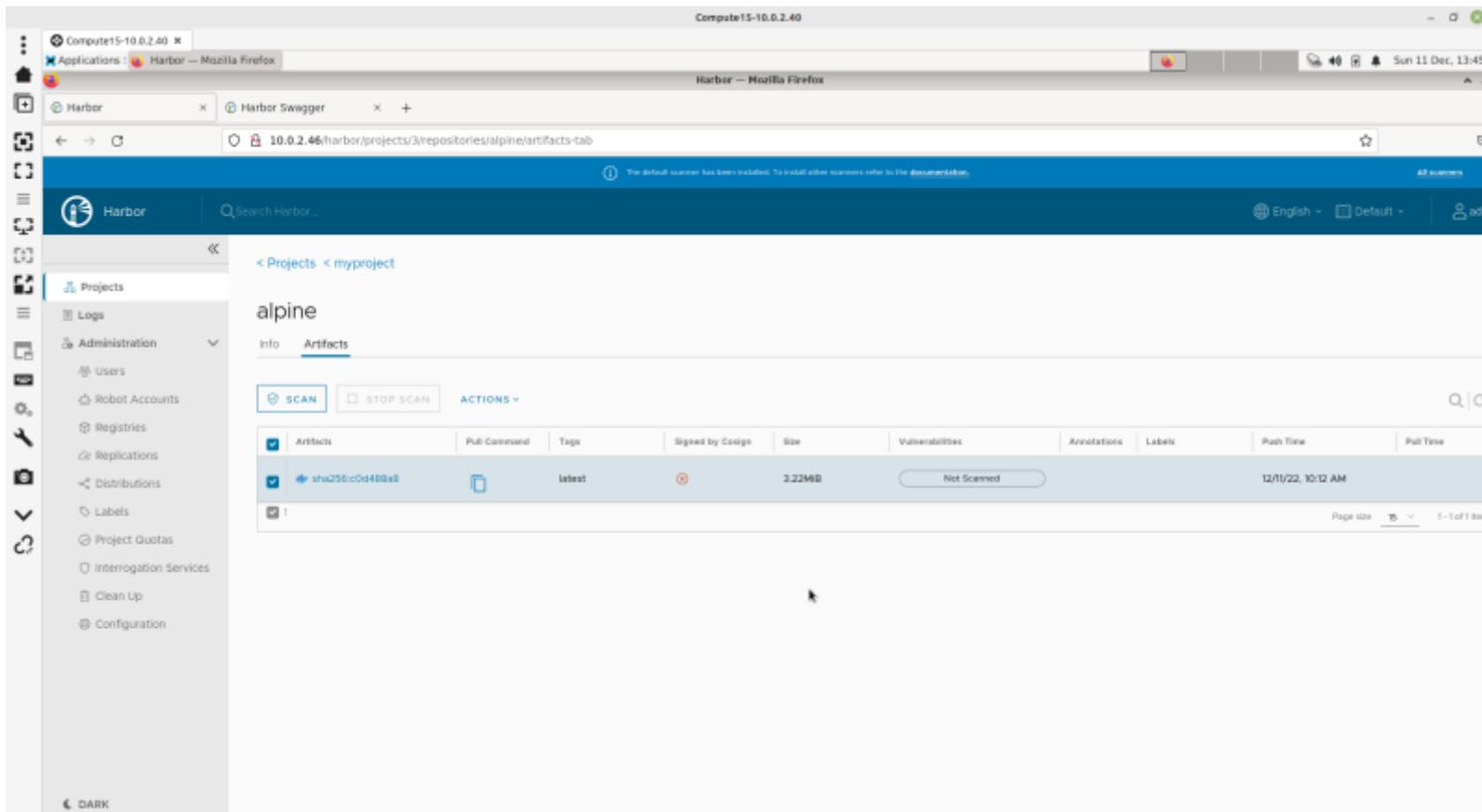


Cliquez sur **project2** > **Labels** > **NEW LABEL** et créez l'étiquette **project2label** :



The screenshot displays the Harbor web interface in a browser window. The address bar shows the URL `10.0.2.46/harbor/projects/5/labels`. The page title is `project2` and the user is `System Admin`. The interface includes a sidebar with navigation options like `Projects`, `Logs`, and `Administration`. The main content area shows the `Labels` tab for `project2`. At the top right, there are boxes for `Access Level` (Public) and `Quota used` (0Byte of unlimited). Below this, there are buttons for `+ NEW LABEL`, `EDIT`, and `DELETE`. A form for creating a new label is visible, with fields for `Label Name` (project2label), `Color` (a dropdown menu), and `Description` (project2label). Below the form is a table with columns `Label`, `Description`, and `Creation Time`. The table is currently empty, and a message `We couldn't find any labels!` is displayed in the center. The bottom right corner shows `Page size 15`.

Cliquez sur **Projects > myproject > myproject/alpine** et sélectionnez l'Artefact :



The screenshot shows the Harbor web interface in a browser window. The URL is `10.0.2.46/harbor/projects/3/repositories/alpine/artifacts-tab`. The page title is `< Projects < myproject`. The main content area shows the `alpine` repository with the `Artifacts` tab selected. Above the table are buttons for `SCAN`, `STOP SCAN`, and `ACTIONS`. The table has the following columns: `Artifacts`, `Pull Command`, `Tags`, `Signed by Cosign`, `Size`, `Vulnerabilities`, `Annotations`, `Labels`, `Push Time`, and `Pull Time`. One artifact is listed with the digest `sha256:0d488a8`, tag `latest`, size `3.22MB`, and a `Not Scanned` button. The `ACTIONS` menu is open, showing `Add Labels` as the first option.

Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
sha256:0d488a8		latest		3.22MB	Not Scanned			12/11/22, 10:12 AM	

Cliquez sur **ACTIONS** > **Add Labels** :

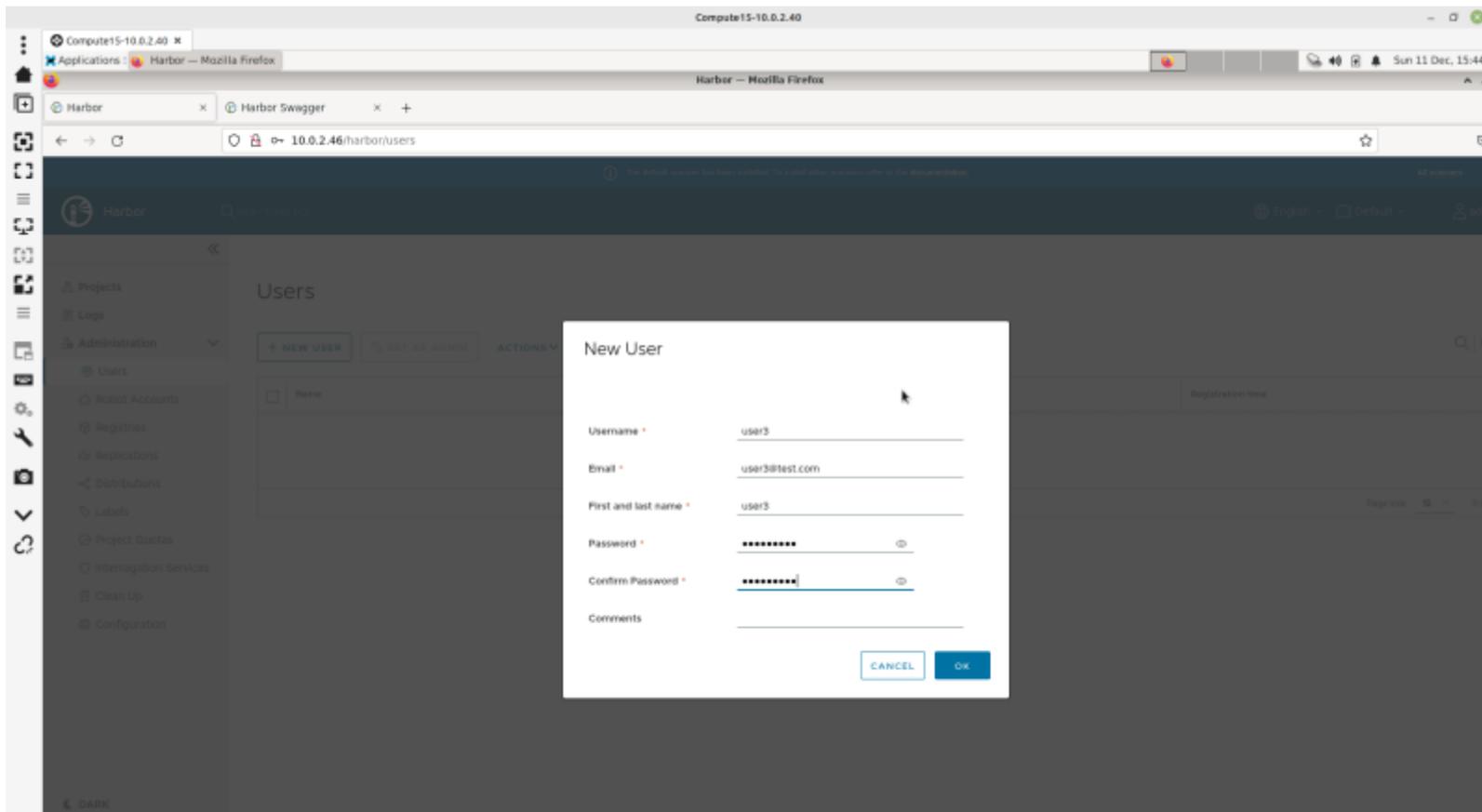
The screenshot shows the Harbor web interface. The main content area displays the 'alpine' repository under the 'myproject' project. The 'Artifacts' tab is selected, showing a table of artifacts. A context menu is open over the first artifact, with the 'Add Labels' option selected. A sub-menu is visible, showing the text 'Add labels to this image' and a search box for labels. The table has columns for Tag, Signed by Cosign, Size, Vulnerabilities, Annotations, Labels, Push Time, and Pull Time. The first artifact has the tag 'sha256:0d488a8' and a size of 3.22MB. The 'Vulnerabilities' column shows 'Not Scanned'. The 'Labels' column is empty. The 'Push Time' is '12/11/22, 10:12 AM'. The 'Pull Time' is empty. The 'Annotations' column is empty. The 'Signed by Cosign' column is empty. The 'Tag' is 'sha256:0d488a8'. The 'Size' is '3.22MB'. The 'Vulnerabilities' is 'Not Scanned'. The 'Annotations' is empty. The 'Labels' is empty. The 'Push Time' is '12/11/22, 10:12 AM'. The 'Pull Time' is empty. The 'Page size' is '25'. The '1-1 of 1 items' is shown at the bottom right of the table.



Important : Notez que l'étiquette **project2label** n'est pas disponible.

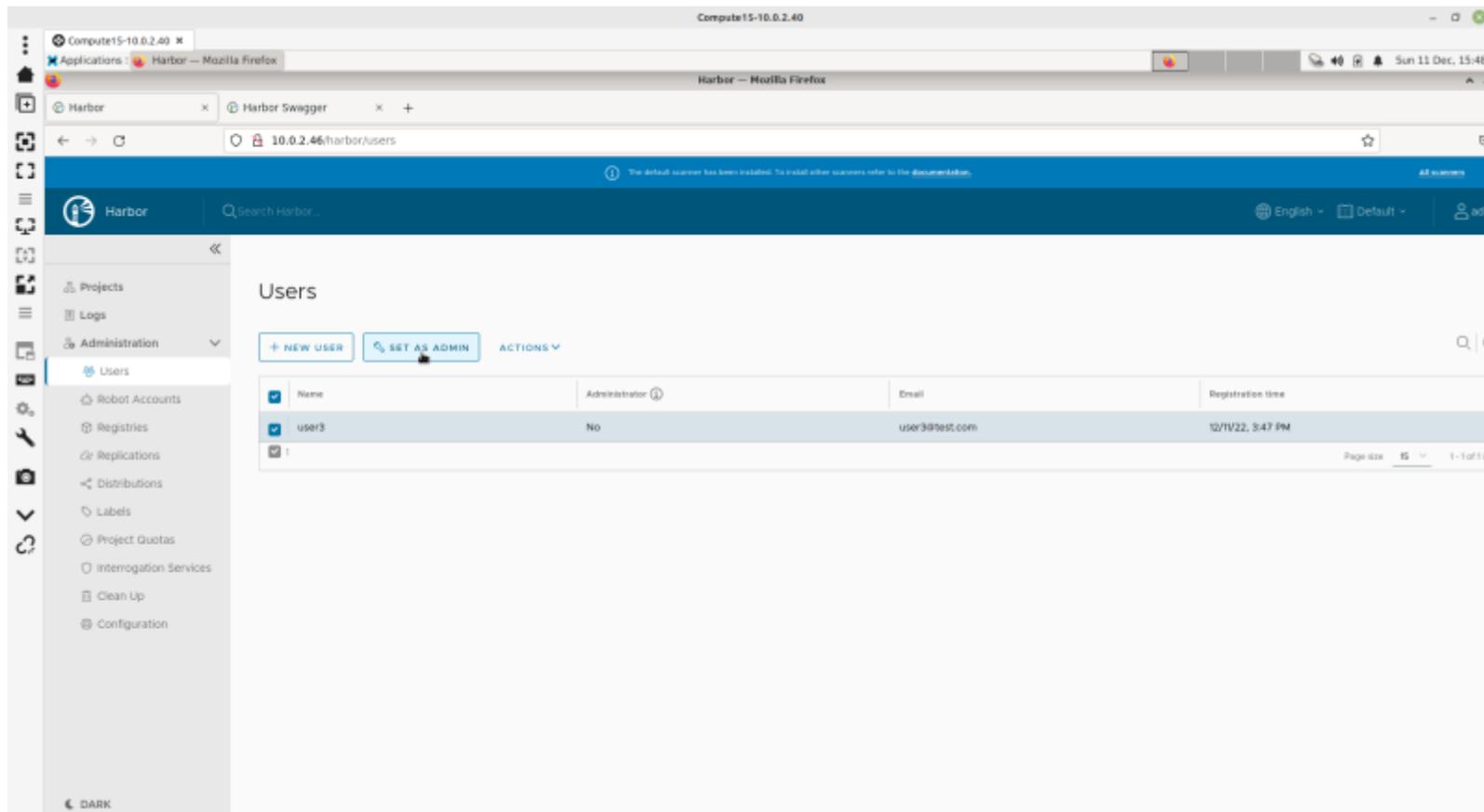
5.9 - Authentification et RBAC

Cliquez sur **Users** puis sur **NEW USER** et créez l'utilisateur **user3** :



Important : Notez que pour être valide le mot de passe doit comporter une lettre Majuscule et un chiffre.

Sélectionnez l'utilisateur **user3** et cliquez sur le bouton **SET AS ADMIN** :

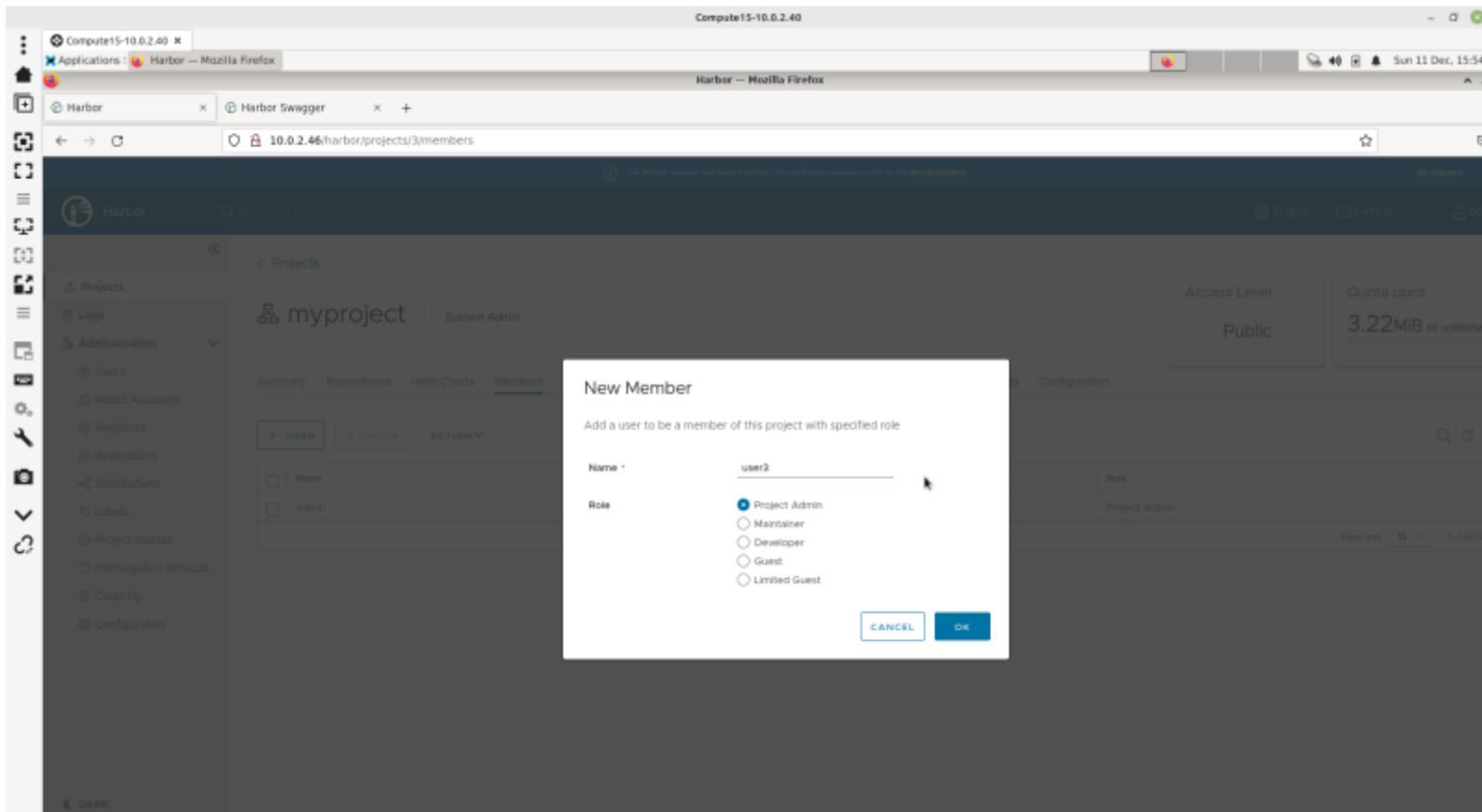


The screenshot shows the Harbor web interface in a browser window. The address bar displays `10.0.2.46/harbor/users`. The main content area is titled "Users" and features a table with the following data:

Name	Administrator	Email	Registration time
user3	No	user3@test.com	12/11/22, 3:47 PM

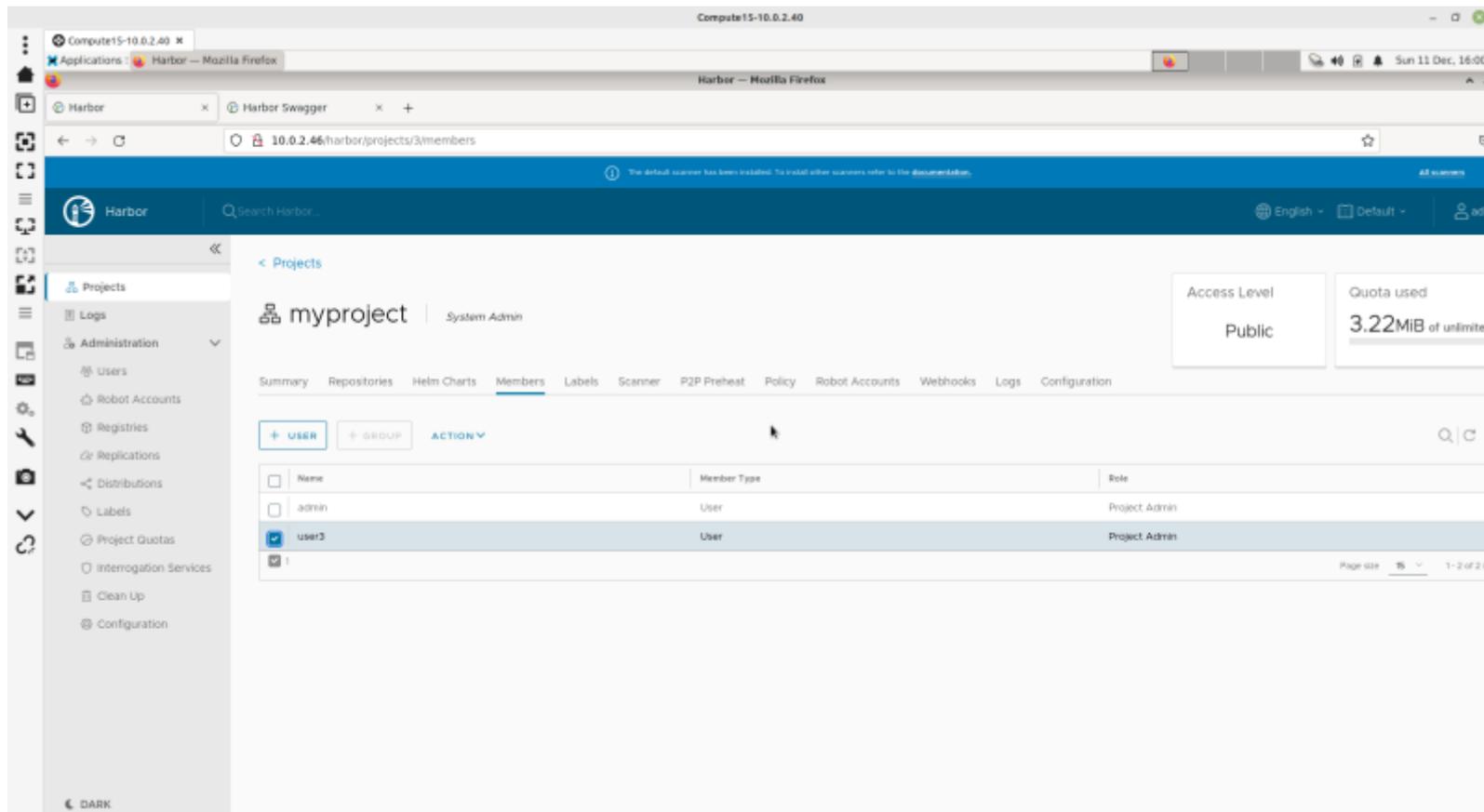
Navigation options include "+ NEW USER" and "SET AS ADMIN". The left sidebar shows the "Administration" menu with "Users" selected. The bottom left corner has a "DARK" theme toggle.

Cliquez sur **Projects > myproject** puis sur **Members** et **+ USER** :



Important : Pour consulter les privilèges de chaque rôle, cliquez sur [ce lien](#).

Cliquez sur **OK** et sélectionnez l'utilisateur **user3** :



The screenshot shows the Harbor web interface for a project named "myproject". The page is titled "Members" and displays a table of project members. The table has three columns: "Name", "Member Type", and "Role". The "user3" member is highlighted, and the "ACTION" dropdown menu is visible above the table.

Name	Member Type	Role
admin	User	Project Admin
user3	User	Project Admin

Pour modifier son rôle, cliquez sur **ACTION** et choisissez le rôle désiré :

The screenshot shows the Harbor web interface for a project named 'myproject'. The 'Members' tab is active, displaying a table of members. A dropdown menu is open over the 'user3' row, showing options: Set Role, Project Admin, Maintainer, Developer, Guest, Limited Guest, and Remove. The 'Maintainer' option is highlighted. The '+ GROUP' button is disabled (greyed out).

Name	Member Type	Role
admin	User	Project Admin
user3	User	Project Admin



Important : Notez que le bouton **+ GROUP** est grisé. Si un Active Directory ou bien LDAP avait été configuré, Harbor pourrait lire la liste des groupes et nous permettrait d'ajouter des utilisateurs à un des groupes.

5.10 - Utilisation d'Images Docker Personnalisées

Créez un répertoire nommé myDocker :

```
root@debian11:~/harbor# cd ..
root@debian11:~# mkdir ~/myDocker
root@debian11:~# cd ~/myDocker
root@debian11:~/myDocker#
```

Créez le fichier myEntrypoint.sh :

```
root@debian11:~/myDocker# vi myEntrypoint.sh
root@debian11:~/myDocker# cat myEntrypoint.sh
#!/bin/bash
if [ -z "$myVariable" ]; then
    echo "La variable myVariable doit être renseignée"
    return 1
fi

while true;
do
    echo $1 \($(date +%H:%M:%S)\);
    sleep "$myVariable";
done
```

Testez ce script :

```
root@debian11:~/myDocker# myVariable=3 . ./myEntrypoint.sh salut
salut (16:12:59)
salut (16:13:02)
salut (16:13:05)
^C
root@debian11:~/myDocker#
```

Rendez ce script exécutable :

```
root@debian11:~/myDocker# chmod u+x myEntrypoint.sh
```

Créez maintenant le fichier **Dockerfile** dans le répertoire **~/myDocker** :

```
root@debian11:~/myDocker# cat Dockerfile
FROM centos:latest
MAINTAINER ittraining "infos@ittraining.io"
COPY myEntrypoint.sh /entrypoint.sh
ENV myVariable 3
ENTRYPOINT ["/entrypoint.sh"]
CMD ["mycommand"]
```

Connectez-vous à Harbor :

```
root@debian11:~/myDocker# docker login 10.0.2.46:80
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Générez maintenant l'image :

```
root@debian11:~/myDocker# docker build --tag "10.0.2.46:80/myproject/mydocker:latest" .
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM centos:latest
latest: Pulling from library/centos
ald0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
---> 5d0da3dc9764
Step 2/6 : MAINTAINER ittraining "infos@ittraining.io"
---> Running in 7e359420ea95
Removing intermediate container 7e359420ea95
---> 7ce8d150d72d
```

```
Step 3/6 : COPY myEntrypoint.sh /entrypoint.sh
---> c1fda8d1e43c
Step 4/6 : ENV myVariable 3
---> Running in edf989144c38
Removing intermediate container edf989144c38
---> 628615e7003a
Step 5/6 : ENTRYPOINT ["/entrypoint.sh"]
---> Running in 0490b6a50bbc
Removing intermediate container 0490b6a50bbc
---> c2ed329815ef
Step 6/6 : CMD ["mycommand"]
---> Running in e2ecfea5be53
Removing intermediate container e2ecfea5be53
---> ccb27534d7f2
Successfully built ccb27534d7f2
Successfully tagged 10.0.2.46:80/myproject/mydocker:latest
```

```
root@debian11:~/myDocker# docker images
```

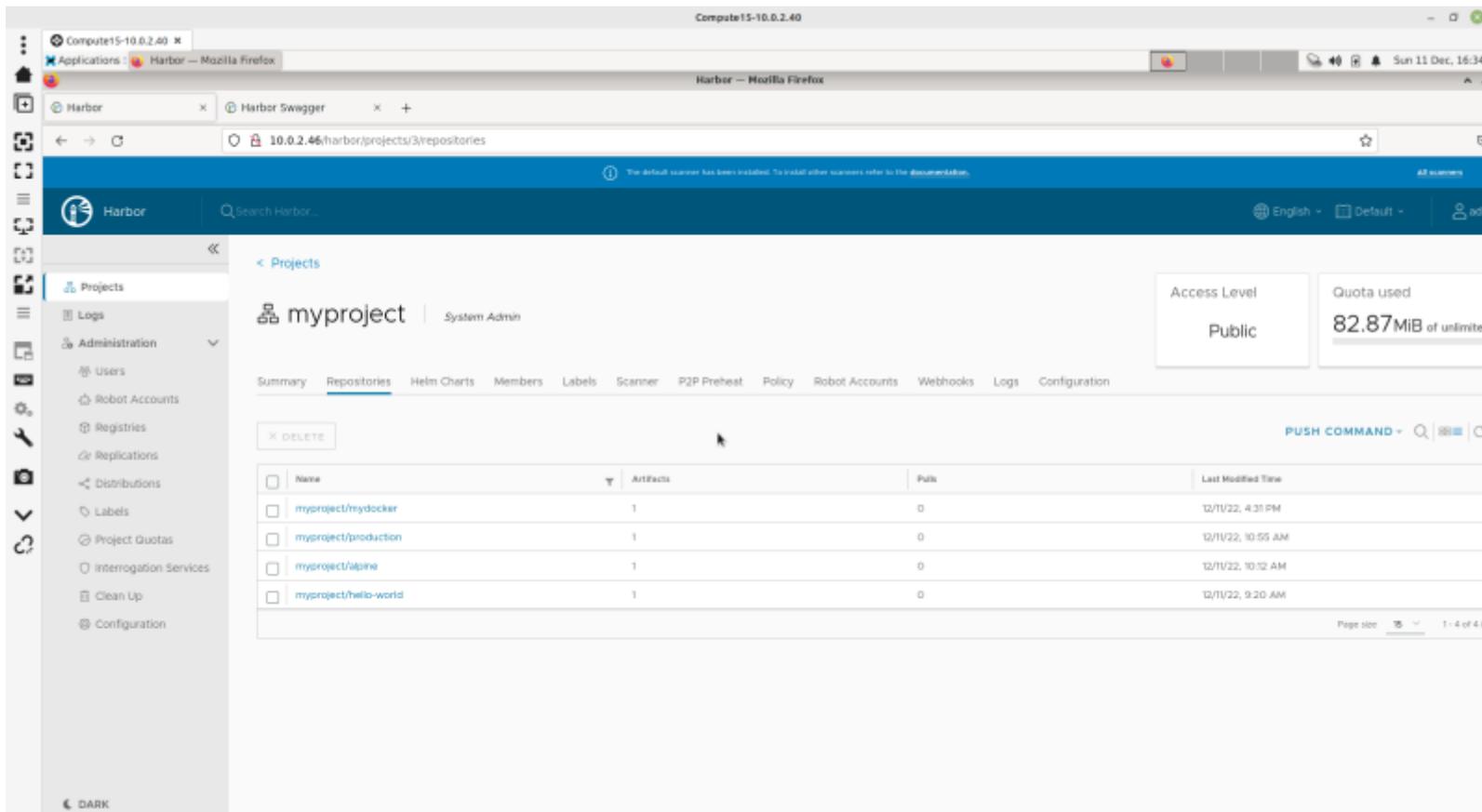
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
10.0.2.46:80/myproject/mydocker	latest	ccb27534d7f2	2 minutes ago	231MB
10.0.2.46:80/myproject/alpine	latest	49176f190c7e	2 weeks ago	7.05MB
alpine	latest	49176f190c7e	2 weeks ago	7.05MB
goharbor/harbor-exporter	v2.6.2	ae230e335853	4 weeks ago	95.4MB
goharbor/chartmuseum-photon	v2.6.2	be361eee8a36	4 weeks ago	225MB
goharbor/redis-photon	v2.6.2	84fa3669bb10	4 weeks ago	154MB
goharbor/trivy-adapter-photon	v2.6.2	beef2527c7b2	4 weeks ago	441MB
goharbor/notary-server-photon	v2.6.2	b43e7c963f06	4 weeks ago	112MB
goharbor/notary-signer-photon	v2.6.2	4e9982286316	4 weeks ago	109MB
goharbor/harbor-registryctl	v2.6.2	d403d4e0f990	4 weeks ago	137MB
goharbor/registry-photon	v2.6.2	596e445b18f0	4 weeks ago	77.5MB
goharbor/nginx-photon	v2.6.2	cbbdf28947a5	4 weeks ago	154MB
goharbor/harbor-log	v2.6.2	3ce902023f73	4 weeks ago	161MB
goharbor/harbor-jobservice	v2.6.2	05791a75c8b8	4 weeks ago	250MB
goharbor/harbor-core	v2.6.2	484e2a64616d	4 weeks ago	213MB

goharbor/harbor-portal	v2.6.2	932297030d4e	4 weeks ago	162MB
goharbor/harbor-db	v2.6.2	351b19e366cb	4 weeks ago	225MB
goharbor/prepare	v2.6.2	eecc993c7643	4 weeks ago	164MB
10.0.2.46:80/myproject/production	latest	feb5d9fea6a5	14 months ago	13.3kB
hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB
10.0.2.46:80/myproject/hello-world	1.0	feb5d9fea6a5	14 months ago	13.3kB
10.0.2.46:80/myproject/hello-world	latest	feb5d9fea6a5	14 months ago	13.3kB
centos	latest	5d0da3dc9764	15 months ago	231MB

Procédez maintenant au **push** :

```
root@debian11:~/myDocker# docker push 10.0.2.46:80/myproject/mydocker:latest
The push refers to repository [10.0.2.46:80/myproject/mydocker]
7c7558b01004: Pushed
74ddd0ec08fa: Pushed
latest: digest: sha256:323ab8f05733ac369963b338bb8fe81bd5cb2d77b46ac3650b7924e7da3aa9cf size: 736
```

Retournez à l'interface de Harbor et cliquez sur **Projects > myproject** :



The screenshot shows the Harbor web interface. The main content area displays the 'myproject' project page. The 'Repositories' tab is selected, showing a table of repositories. The table has columns for 'Name', 'Artifacts', 'Push', and 'Last Modified Time'. The first row, 'myproject/mydocker', is highlighted. Other repositories listed are 'myproject/production', 'myproject/alpine', and 'myproject/hello-world'. The interface also shows a sidebar with navigation options like 'Projects', 'Logs', 'Administration', 'Users', 'Robot Accounts', 'Registries', 'Replications', 'Distributions', 'Labels', 'Project Quotas', 'Interrogation Services', 'Clean Up', and 'Configuration'. The top right corner shows 'Access Level: Public' and 'Quota used: 82.87 MiB of unlimited'.

Name	Artifacts	Push	Last Modified Time
myproject/mydocker	1	0	12/11/22, 4:31 PM
myproject/production	1	0	12/11/22, 10:55 AM
myproject/alpine	1	0	12/11/22, 10:12 AM
myproject/hello-world	1	0	12/11/22, 9:20 AM



Important : Notez que le dépôt **myproject/mydocker** a été créé.

Cliquez sur **myproject/mydocker** :

The screenshot shows the Harbor web interface. The main content area displays the 'mydocker' project page, specifically the 'Artifacts' tab. A table lists the artifacts, with one entry: 'sha256:323ab810' with a 'latest' tag and a size of 79.65MB. The artifact is marked as 'Not Scanned'. The interface includes a sidebar with navigation options like 'Projects', 'Logs', and 'Administration', and a top navigation bar with 'Harbor' and 'myproject' breadcrumbs.



Important : Notez que l'Artefact est présent et qu'il possède un tag **latest**.

Refaites un build de l'image en changeant le tag :

```
root@debian11:~/myDocker# docker build --tag "10.0.2.46:80/myproject/mydocker:rel1" .
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM centos:latest
--> 5d0da3dc9764
Step 2/6 : MAINTAINER ittraining "infos@ittraining.io"
```

```
---> Using cache
---> 7ce8d150d72d
Step 3/6 : COPY myEntrypoint.sh /entrypoint.sh
---> Using cache
---> c1fda8d1e43c
Step 4/6 : ENV myVariable 3
---> Using cache
---> 628615e7003a
Step 5/6 : ENTRYPOINT ["/entrypoint.sh"]
---> Using cache
---> c2ed329815ef
Step 6/6 : CMD ["mycommand"]
---> Using cache
---> ccb27534d7f2
Successfully built ccb27534d7f2
Successfully tagged 10.0.2.46:80/myproject/mydocker:rel1
```



Important : Notez l'utilisation des images intermédiaires issues du cache.

Procédez maintenant au **push** :

```
root@debian11:~/myDocker# docker push 10.0.2.46:80/myproject/mydocker:rel1
The push refers to repository [10.0.2.46:80/myproject/mydocker]
7c7558b01004: Layer already exists
74ddd0ec08fa: Layer already exists
rel1: digest: sha256:323ab8f05733ac369963b338bb8fe81bd5cb2d77b46ac3650b7924e7da3aa9cf size: 736
```



Important : Notez que seul le nouveau tag a été poussé.

Editez le Dockerfile et changez la valeur de **myVariable** de 3 à 5 :

```
root@debian11:~/myDocker# vi Dockerfile
root@debian11:~/myDocker# cat Dockerfile
FROM centos:latest
MAINTAINER ittraining "infos@ittraining.io"
COPY myEntrypoint.sh /entrypoint.sh
ENV myVariable 5
ENTRYPOINT ["/entrypoint.sh"]
CMD ["mycommand"]
```

Compiliez de nouveau l'image :

```
root@debian11:~/myDocker# docker build --tag "10.0.2.46:80/myproject/mydocker:rel2" .
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM centos:latest
---> 5d0da3dc9764
Step 2/6 : MAINTAINER ittraining "infos@ittraining.io"
---> Using cache
---> 7ce8d150d72d
Step 3/6 : COPY myEntrypoint.sh /entrypoint.sh
---> Using cache
---> c1fda8d1e43c
Step 4/6 : ENV myVariable 5
---> Running in c90f60beee6f
Removing intermediate container c90f60beee6f
---> d9a704d55c69
Step 5/6 : ENTRYPOINT ["/entrypoint.sh"]
---> Running in 8aabb88b172d
Removing intermediate container 8aabb88b172d
---> 6ca7239b3489
Step 6/6 : CMD ["mycommand"]
---> Running in 8e1bd939a7c2
Removing intermediate container 8e1bd939a7c2
---> 9a12b543b94b
Successfully built 9a12b543b94b
```

Successfully tagged 10.0.2.46:80/myproject/mydocker:rel2

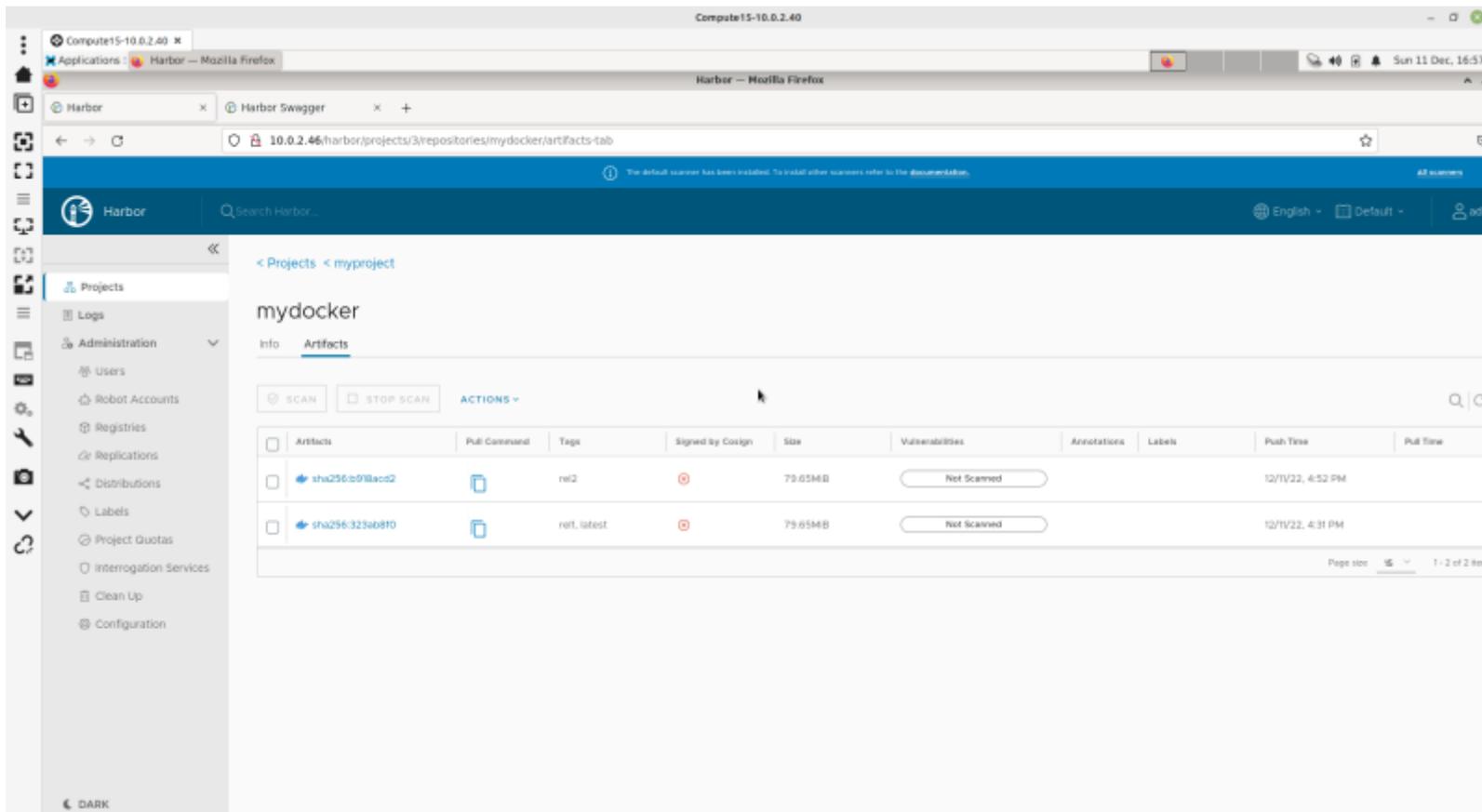
Procédez maintenant au **push** :

```
root@debian11:~/myDocker# docker push 10.0.2.46:80/myproject/mydocker:rel2
The push refers to repository [10.0.2.46:80/myproject/mydocker]
7c7558b01004: Layer already exists
74ddd0ec08fa: Layer already exists
rel2: digest: sha256:b918acd244c70cf528d8e96584d8d34099d2a8eefc1edd86ba972dd975934f52 size: 736
```



Important : Notez que la valeur du SHA256 de rel2 est différente - **b918acd244c70cf528d8e96584d8d34099d2a8eefc1edd86ba972dd975934f52** par rapport à **323ab8f05733ac369963b338bb8fe81bd5cb2d77b46ac3650b7924e7da3aa9cf**.

De ce fait un nouvel Artefact a été créé :



Editez le Dockerfile et changez la valeur de **CMD** de "mycommand" à "**mycommand modified**" :

```
root@debian11:~/myDocker# cat Dockerfile
FROM centos:latest
MAINTAINER ittraining "infos@ittraining.io"
COPY myEntrypoint.sh /entrypoint.sh
ENV myVariable 5
ENTRYPOINT ["/entrypoint.sh"]
CMD ["mycommand modified"]
```

Compiliez de nouveau l'image en utilisant le même tag qu'avant :

```
root@debian11:~/myDocker# docker build --tag "10.0.2.46:80/myproject/mydocker:rel2" .
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM centos:latest
---> 5d0da3dc9764
Step 2/6 : MAINTAINER ittraining "infos@ittraining.io"
---> Using cache
---> 7ce8d150d72d
Step 3/6 : COPY myEntrypoint.sh /entrypoint.sh
---> Using cache
---> c1fda8d1e43c
Step 4/6 : ENV myVariable 5
---> Using cache
---> d9a704d55c69
Step 5/6 : ENTRYPOINT ["/entrypoint.sh"]
---> Using cache
---> 6ca7239b3489
Step 6/6 : CMD ["mycommand modified"]
---> Running in ac174f91145b
Removing intermediate container ac174f91145b
---> 56cc6362dc99
Successfully built 56cc6362dc99
Successfully tagged 10.0.2.46:80/myproject/mydocker:rel2
```

Procédez maintenant au **push** :

```
root@debian11:~/myDocker# docker push 10.0.2.46:80/myproject/mydocker:rel2
The push refers to repository [10.0.2.46:80/myproject/mydocker]
7c7558b01004: Layer already exists
74ddd0ec08fa: Layer already exists
rel2: digest: sha256:fcf710f98e14ac3df4d3cf365f04423da8ceb42325eca9efe87f67082904728e size: 736
```



Important : Notez que la valeur du SHA256 de rel2 est différente -



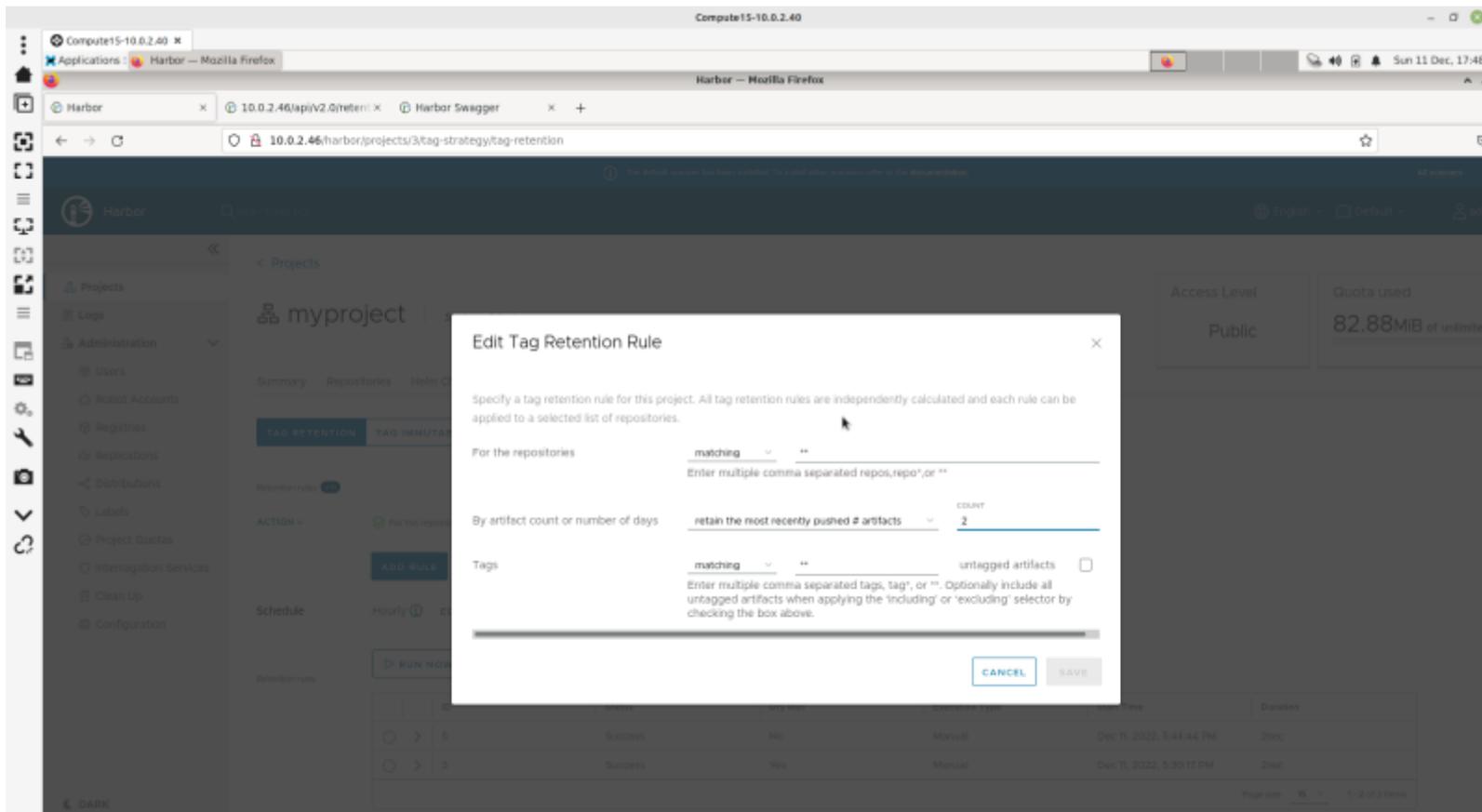
fcf710f98e14ac3df4d3cf365f04423da8ceb42325eca9efe87f67082904728e par rapport à **b918acd244c70cf528d8e96584d8d34099d2a8eefc1edd86ba972dd975934f52**. Par contre, puisque le même tag a été utilisé, le tag **rel2** a été attribué au dernier Artefact créé et le même tag a été supprimé de l'Artefact dont le SHA256 est **b918acd244c70cf528d8e96584d8d34099d2a8eefc1edd86ba972dd975934f52**

5.11 - Rétention de Tags

Pour gérer ce comportement, il est possible de créer des règles. Cliquez sur **myproject > Policy** :

The screenshot shows the Harbor web interface in a browser window. The page is titled 'myproject' and is accessed by 'System Admin'. The navigation menu on the left includes 'Projects', 'Logs', 'Administration', 'Users', 'Robot Accounts', 'Registries', 'Replications', 'Distributions', 'Labels', 'Project Quotas', 'Interrogation Services', 'Clean Up', and 'Configuration'. The main content area is for 'TAG RETENTION' configuration. It features a 'Retention rules' section with an 'ADD RULE' button. Below this is a 'Schedule' section with a 'None' dropdown and an 'EDIT' link. At the bottom, there are buttons for 'RUN NOW', 'DRY RUN', 'ABORT', and a refresh icon. A table with columns 'ID', 'Status', 'Dry Run', 'Execution Type', 'Start Time', and 'Duration' is shown, but it is empty with a message: 'We couldn't find any executions!'. The top right of the page shows 'Access Level: Public' and 'Quota used: 82.88MiB of unlimited'. The browser address bar shows '10.0.2.46/harbor/projects/3/tag-strategy/tag-retention'.

Cliquez sur **ADD RULE** et remplissez le formulaire ainsi :



Important : Cette règle s'appliquera à tous les dépôts et concernera tous les tags. La règle retiendra les 2 Artefacts les plus récemment poussés vers le dépôt.

Cliquez sur **EDIT** sur la ligne **Schedule** et choisissez **hourly** puis cliquez **Save** :

The screenshot shows the Harbor web interface in a browser window. The page is titled 'myproject' and is accessed by 'System Admin'. The main content area is for 'TAG RETENTION' configuration. A retention rule is defined with the following details:

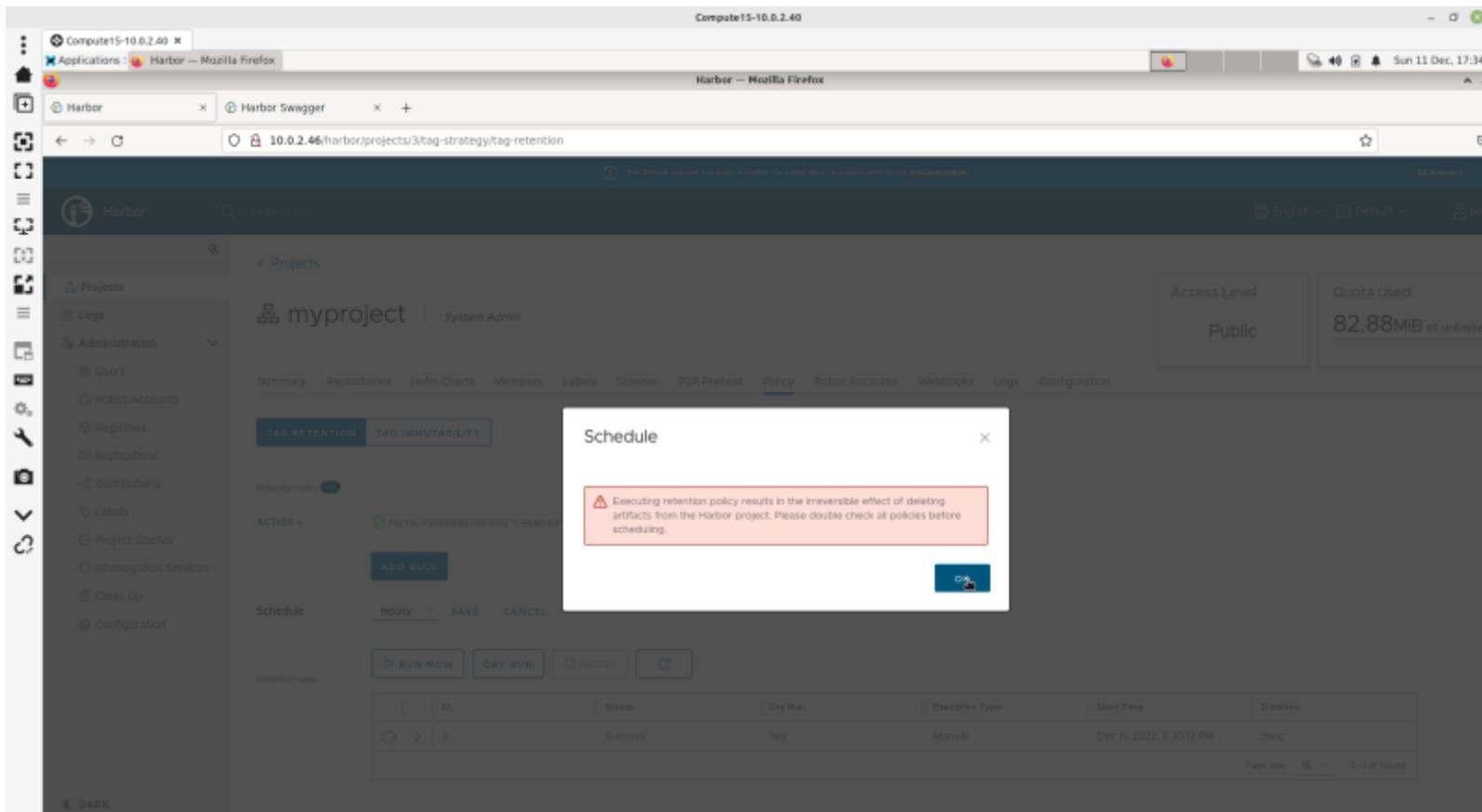
- Retention rules:** 1 rule active.
- ACTION:** For the repositories matching "", retain the most recently pushed 2 artifacts with tags matching "".
- Schedule:** Hourly.
- Buttons:** ADD RULE, RUN NOW, DRY RUN, ABORT, and a refresh icon.

Below the configuration, a table shows the execution history of retention runs:

ID	Status	Dry Run	Execution Type	Start Time	Duration
3	Success	Yes	Manual	Dec 11, 2022, 5:30:17 PM	2sec

At the bottom right of the table, it indicates 'Page size: 15' and '1-1 of 1 items'. A 'DARK' theme toggle is visible in the bottom left corner of the interface.

Cliquez sur **OK** dans la boîte d'avertissement :



Cliquez ensuite sur le bouton **DRY RUN**. Quand le Dry Run a terminé avec le message **Success**, cliquez sur le bouton **Expand** du Dry Run :

The screenshot shows the Harbor web interface in a browser window. The page is titled 'myproject' and is in the 'System Admin' view. The 'Policy' tab is selected, and the 'TAG RETENTION' sub-tab is active. The retention rule is configured to retain the most recently pushed 2 artifacts with tags matching a pattern. The schedule is set to 'Hourly'. A table of retention runs shows one successful run on Dec 11, 2022, at 5:30:17 PM, with a duration of 2 seconds. The 'Log' button is highlighted on the last row of the table.

Access Level: Public
Quota used: 82.88MiB of unlimited

Retention rules: 1

ACTION: For the repositories matching "", retain the most recently pushed 2 artifacts with tags matching ""

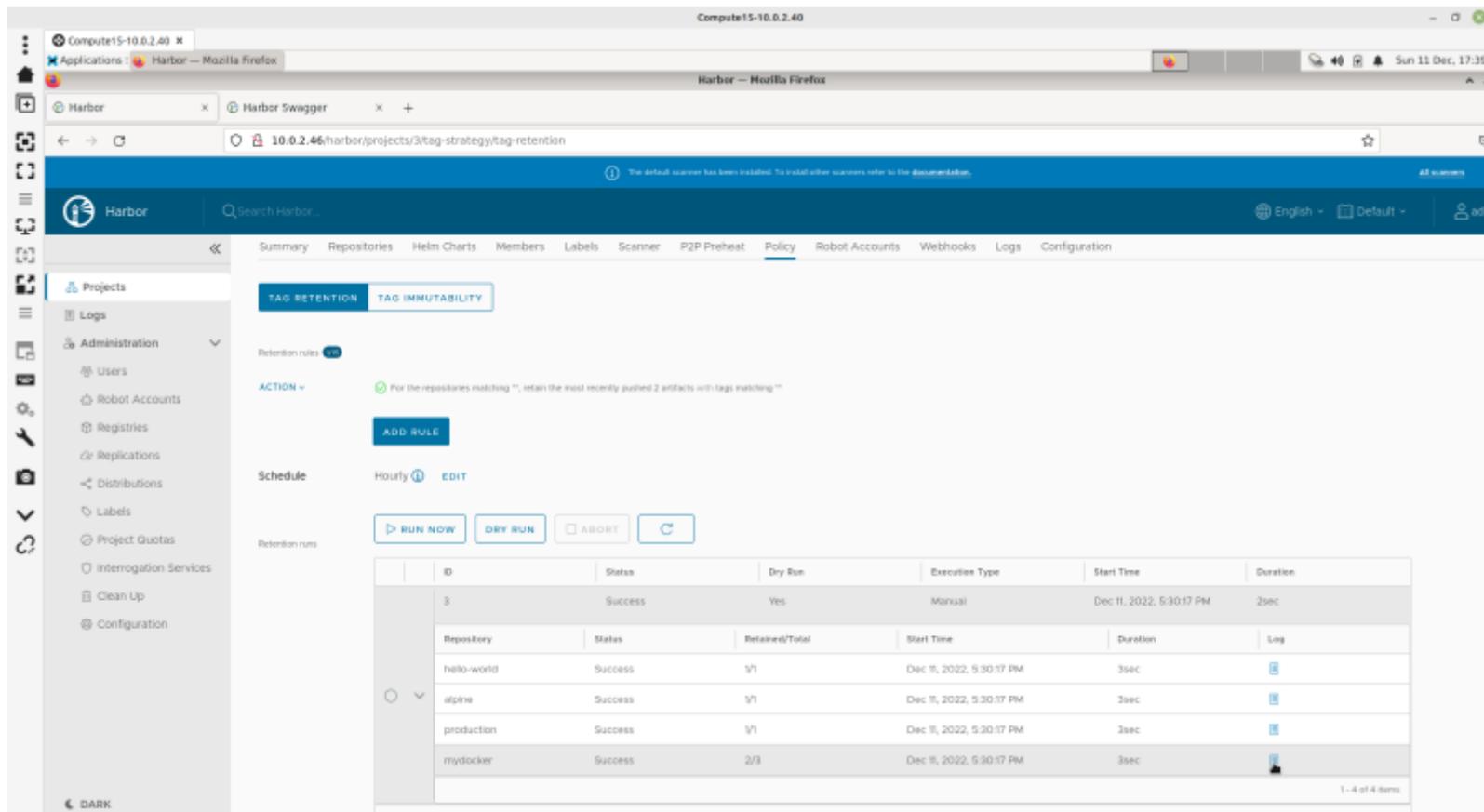
Schedule: Hourly EDIT

Retention runs

ID	Status	Dry Run	Execution Type	Start Time	Duration
3	Success	Yes	Manual	Dec 11, 2022, 5:30:17 PM	2sec

Page size: 15 1-1 of 1 items

Sur la dernière ligne, cliquez sur le bouton **Log** :



The screenshot shows the Harbor web interface for configuring Tag Retention. The main content area displays the following configuration:

- Retention rules:** 1
- ACTION:** For the repositories matching "", retain the most recently pushed 2 artifacts with tags matching ""
- Schedule:** Hourly
- Buttons:** ADD RULE, RUN NOW, DRY RUN, ABORT, Refresh

Below the configuration, a table shows the results of a retention run:

ID	Status	Dry Run	Execution Type	Start Time	Duration
3	Success	Yes	Manual	Dec 11, 2022, 5:30:17 PM	2sec
Repository	Status	Retained/Total	Start Time	Duration	Log
hello-world	Success	1/1	Dec 11, 2022, 5:30:17 PM	3sec	Log
alpine	Success	1/1	Dec 11, 2022, 5:30:17 PM	3sec	Log
production	Success	1/1	Dec 11, 2022, 5:30:17 PM	3sec	Log
mydocker	Success	2/3	Dec 11, 2022, 5:30:17 PM	3sec	Log

1 - 4 of 4 items

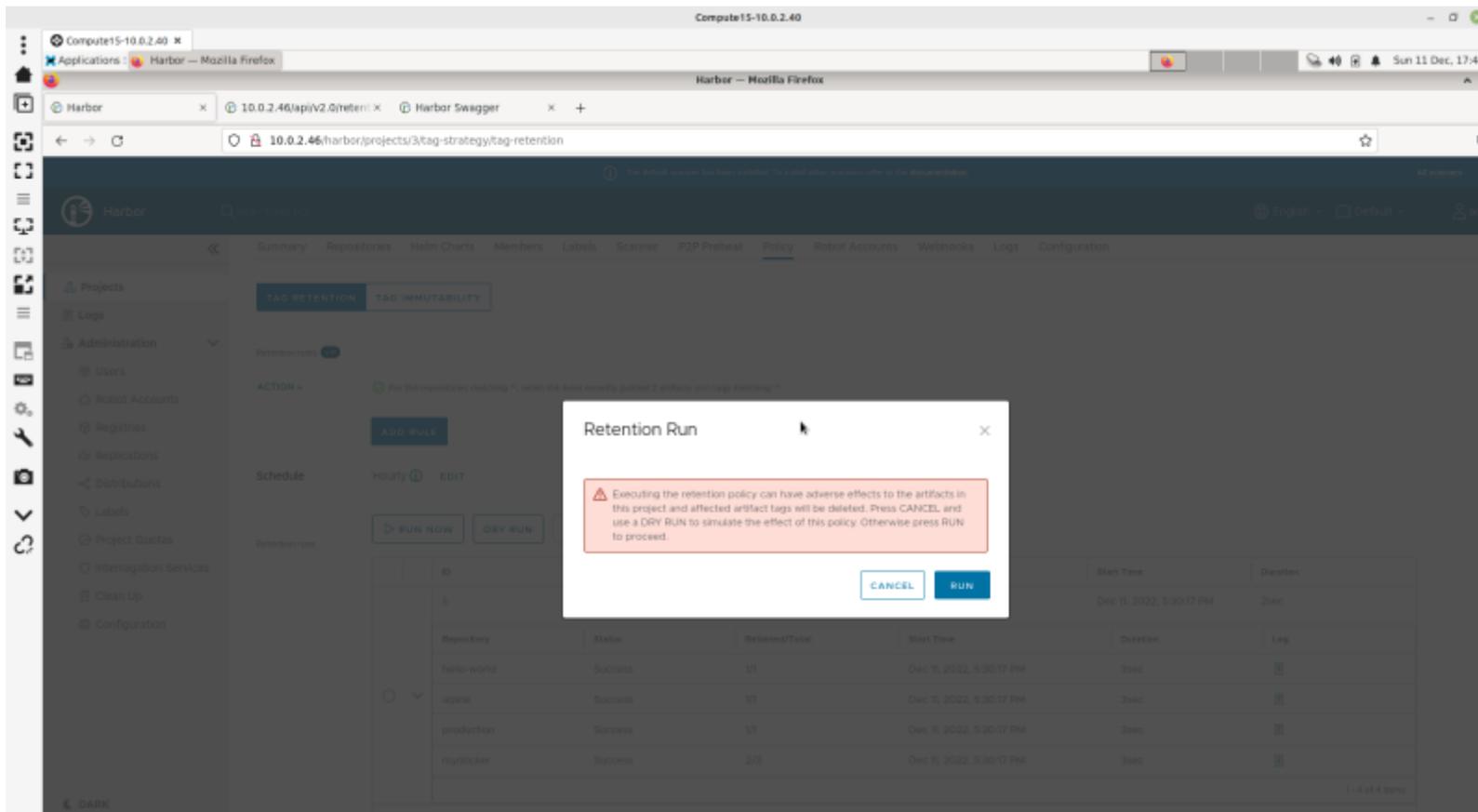
Consultez la colonne **Retention** :



```
2022-12-11T20:30:20Z [INFO] [/pkg/retentier/job.go:86]: Run retention process.
Repository: myproject/mydocker
Rule Algorithm: or
Dry Run: true
2022-12-11T20:30:20Z [INFO] [/pkg/retentier/job.go:101]: Lead 3 candidates from repository myproject/mydocker
2022-12-11T20:30:20Z [INFO] [/pkg/retentier/job.go:212]:
+-----+-----+-----+-----+-----+-----+-----+-----+
| Digest | Tag | Kind | Labels | PushedTime | PulledTime | CreatedTime | Retention |
+-----+-----+-----+-----+-----+-----+-----+-----+
| sha256:fcf718f98e04ac36f483cf385f044236a8ceb42325ecabef87f67082904728e | rel2 | image | | 2022/12/11 16:09:06 | | 2022/12/11 16:09:06 | RETAIN |
| sha256:8938acd244c78cf52888e9658468d34809d2a8eefc1edd88ba072dd975934f52 | | | | 2022/12/11 15:52:42 | | 2022/12/11 15:52:42 | DEL |
| sha256:323ab8f85733ac369963b378bb8fe81b55cb2d77b46ac3658b7924e7da3aa9cf | rel1,latest | image | | 2022/12/11 15:42:28 | | 2022/12/11 15:31:57 | RETAIN |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Revenez dans l'interface de Harbor et exécutez la règle en cliquant sur le bouton **RUN NOW**, puis cliquez sur **RUN** dans la boîte d'avertissement :



Editez de nouveau le Dockerfile et changez la valeur de **myVariable** de 5 à **10** :

```
root@debian11:~/myDocker# vi Dockerfile
root@debian11:~/myDocker# cat Dockerfile
FROM centos:latest
MAINTAINER ittraining "infos@ittraining.io"
COPY myEntrypoint.sh /entrypoint.sh
ENV myVariable 10
ENTRYPOINT ["/entrypoint.sh"]
CMD ["mycommand"]
```

Compilez de nouveau l'image :

```
root@debian11:~/myDocker# docker build --tag "10.0.2.46:80/myproject/mydocker:rel3" .
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM centos:latest
----> 5d0da3dc9764
Step 2/6 : MAINTAINER ittraining "infos@ittraining.io"
----> Using cache
----> 7ce8d150d72d
Step 3/6 : COPY myEntrypoint.sh /entrypoint.sh
----> Using cache
----> c1fda8d1e43c
Step 4/6 : ENV myVariable 10
----> Running in 7c938717175f
Removing intermediate container 7c938717175f
----> ef07118050c5
Step 5/6 : ENTRYPOINT ["/entrypoint.sh"]
----> Running in b2dc04a9363d
Removing intermediate container b2dc04a9363d
----> f73a3c2f7930
Step 6/6 : CMD ["mycommand modified"]
----> Running in fd7a8952dd7f
Removing intermediate container fd7a8952dd7f
----> 3bbe34d8d430
Successfully built 3bbe34d8d430
Successfully tagged 10.0.2.46:80/myproject/mydocker:rel3
```

Procédez maintenant au **push** :

```
root@debian11:~/myDocker# docker push 10.0.2.46:80/myproject/mydocker:rel3
The push refers to repository [10.0.2.46:80/myproject/mydocker]
7c7558b01004: Layer already exists
74ddd0ec08fa: Layer already exists
```

```
rel3: digest: sha256:ae649a586f5d9dcae71440bf88eb98d95c3cc86d40f27eb54fa68e65869c3b48 size: 736
```

Exécutez la règle :

The screenshot shows the Harbor web interface for a project named 'myproject'. The 'TAG RETENTION' tab is active, displaying a retention rule. The rule is configured to retain the most recently pushed 2 artifacts with tags matching '*'. The schedule is set to 'Hourly'. The retention runs table shows three successful runs.

ID	Status	Dry Run	Execution Type	Start Time	Duration
6	Success	No	Scheduled	Dec 11, 2022, 6:00:02 PM	0
5	Success	No	Manual	Dec 11, 2022, 5:44:44 PM	2sec
3	Success	Yes	Manual	Dec 11, 2022, 5:30:17 PM	2sec

Constatez la liste des Artefacts dans **mydocker** :

The screenshot shows the Harbor web interface for the 'myproject' repository. The 'Artifacts' tab is active, displaying a table of artifacts. The table has the following columns: Artifacts, Pull Command, Tags, Signed by Cosign, Size, Vulnerabilities, Annotations, Labels, Push Time, and Pull Time. Two artifacts are listed:

Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
sha256:a6643a5d	[Icon]	ret3	[Red Circle with Exclamation Mark]	79.65MiB	Not Scanned			12/11/22, 6:05 PM	
sha256:c7710f9	[Icon]	ret2	[Red Circle with Exclamation Mark]	79.65MiB	Not Scanned			12/11/22, 5:06 PM	

The 'Signed by Cosign' column shows a red circle with a white exclamation mark for both artifacts, indicating a signing error. The 'Vulnerabilities' column shows 'Not Scanned' for both artifacts. The 'Push Time' for the first artifact is '12/11/22, 6:05 PM' and for the second is '12/11/22, 5:06 PM'. The interface includes a sidebar with navigation options like Projects, Logs, Administration, Users, Robot Accounts, Registries, Replications, Distributions, Labels, Project Quotas, Interrogation Services, Clean Up, and Configuration. The top navigation bar shows the Harbor logo, search bar, and user profile.



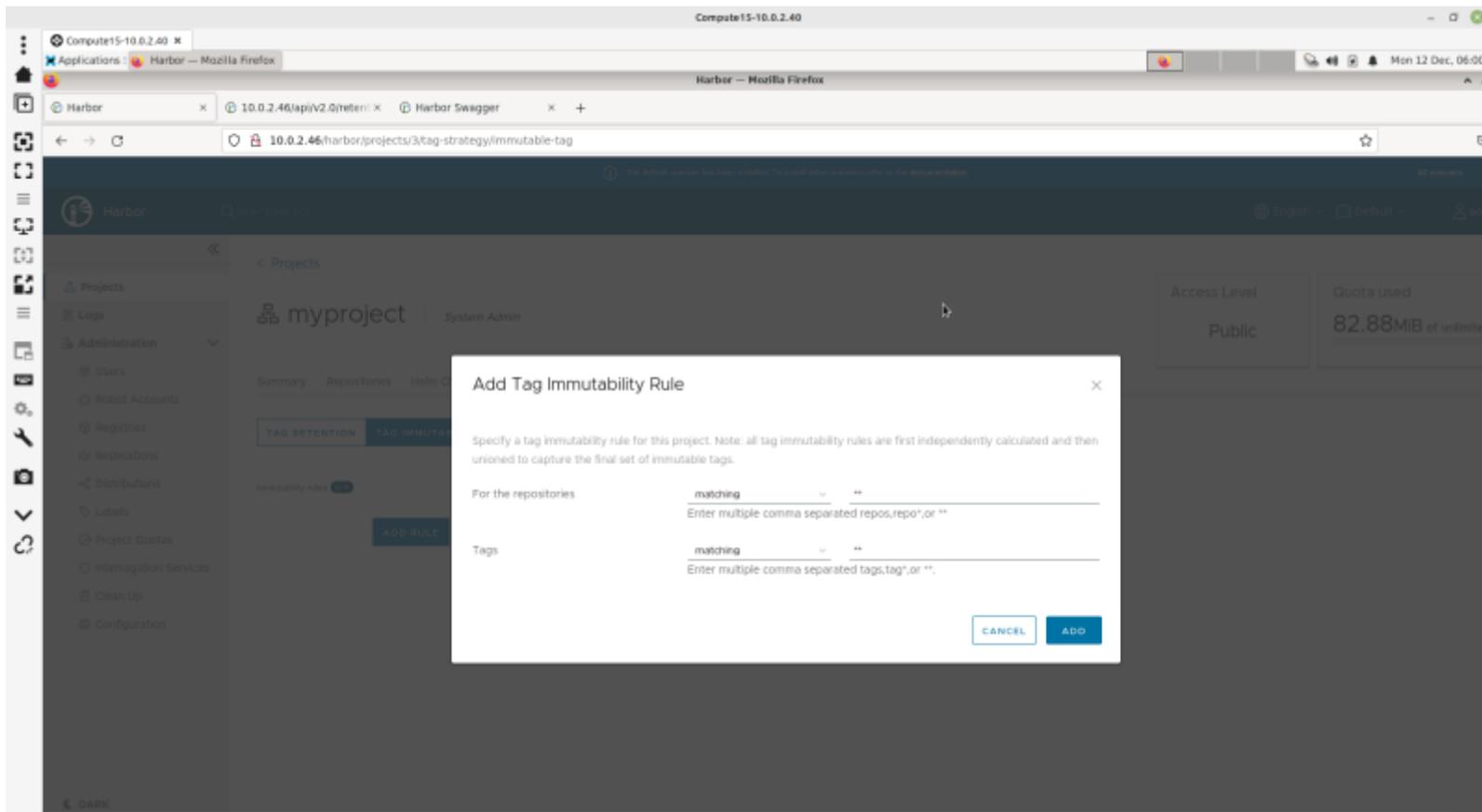
Important : Notez que c'est l'Artefact le plus ancien qui a été supprimé.

5.12 - Immutabilité des Tags

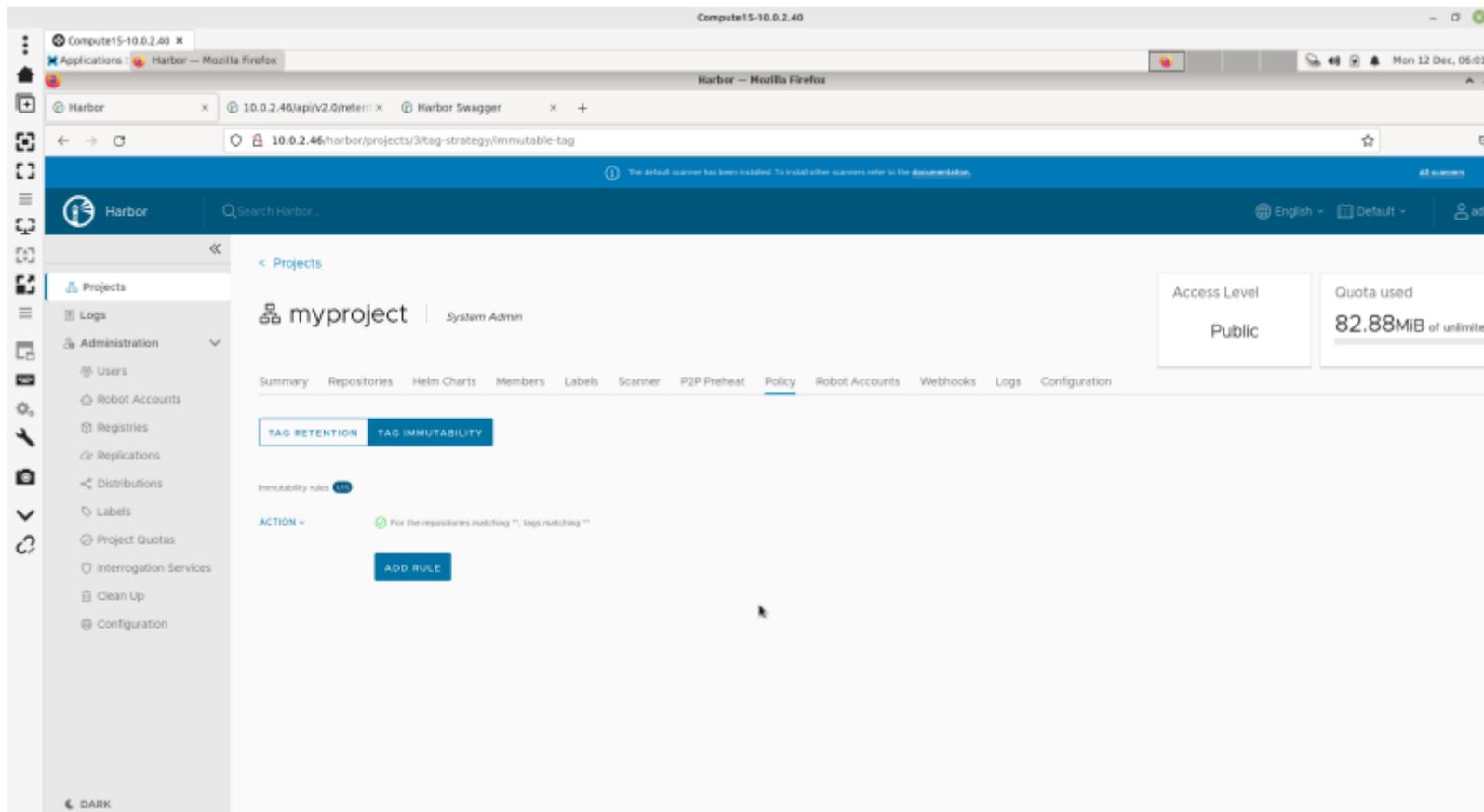
Cliquez sur **myproject** > **Policy** > **TAG IMMUTABILITY** :

The screenshot displays the Harbor web interface. The browser window shows the URL `10.0.2.46/harbor/projects/3/tag-strategy/immutable-tag`. The interface includes a sidebar with navigation options such as 'Projects', 'Logs', 'Administration', 'Users', 'Robot Accounts', 'Registries', 'Replications', 'Distributions', 'Labels', 'Project Quotas', 'Interrogation Services', 'Clean Up', and 'Configuration'. The main content area is titled 'myproject' and shows the 'Policy' tab selected. The 'TAG IMMUTABILITY' section is active, displaying 'Inmutability rules' and a blue 'ADD RULE' button. The 'Access Level' is set to 'Public' and the 'Quota used' is '82.88MiB of unlimited'. The interface also shows a search bar and a 'DARK' theme toggle at the bottom left.

Cliquez maintenant sur le bouton **ADD RULE** :



Cliquez sur le bouton **ADD** :



The screenshot shows the Harbor web interface for a project named 'myproject'. The 'TAG IMMUTABILITY' tab is active, showing a rule configuration. The rule is set to 'Public' access level and '82.88MiB of unlimited' quota used. The rule text is 'For the repositories matching "", tags matching ""'. There is an 'ADD RULE' button below the rule.



Important : Cette règle s'appliquera à tous les dépôts et concernera tous les tags.

Essayez de faire un push de l'image **mydocker:rel2** :

```
root@debian11:~/myDocker# docker push 10.0.2.46:80/myproject/mydocker:rel2
The push refers to repository [10.0.2.46:80/myproject/mydocker]
7c7558b01004: Layer already exists
74ddd0ec08fa: Layer already exists
```

unknown: Failed to process request due to 'mydocker:rel2' configured as immutable.



Important : Notez qu'il n'est pas possible de faire un push car la règle d'immuabilité l'interdit.

Editez le Dockerfile et changez la valeur de **myVariable** de 10 à **8** :

```
root@debian11:~/myDocker# vi Dockerfile
root@debian11:~/myDocker# cat Dockerfile
FROM centos:latest
MAINTAINER ittraining "infos@ittraining.io"
COPY myEntrypoint.sh /entrypoint.sh
ENV myVariable 8
ENTRYPOINT ["/entrypoint.sh"]
CMD ["mycommand"]
```

Faites un build de **mydocker:rel4** :

```
root@debian11:~/myDocker# docker build --tag "10.0.2.46:80/myproject/mydocker:rel4" .
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM centos:latest
---> 5d0da3dc9764
Step 2/6 : MAINTAINER ittraining "infos@ittraining.io"
---> Using cache
---> 7ce8d150d72d
Step 3/6 : COPY myEntrypoint.sh /entrypoint.sh
---> Using cache
---> c1fda8d1e43c
Step 4/6 : ENV myVariable 8
---> Running in 541758cf561f
Removing intermediate container 541758cf561f
```

```
---> 45f622ff8362
Step 5/6 : ENTRYPOINT ["/entrypoint.sh"]
---> Running in 9d309f201613
Removing intermediate container 9d309f201613
---> 749b47fe344a
Step 6/6 : CMD ["mycommand modified"]
---> Running in e7397e408526
Removing intermediate container e7397e408526
---> 85f1cbc19cee
Successfully built 85f1cbc19cee
Successfully tagged 10.0.2.46:80/myproject/mydocker:rel4
```

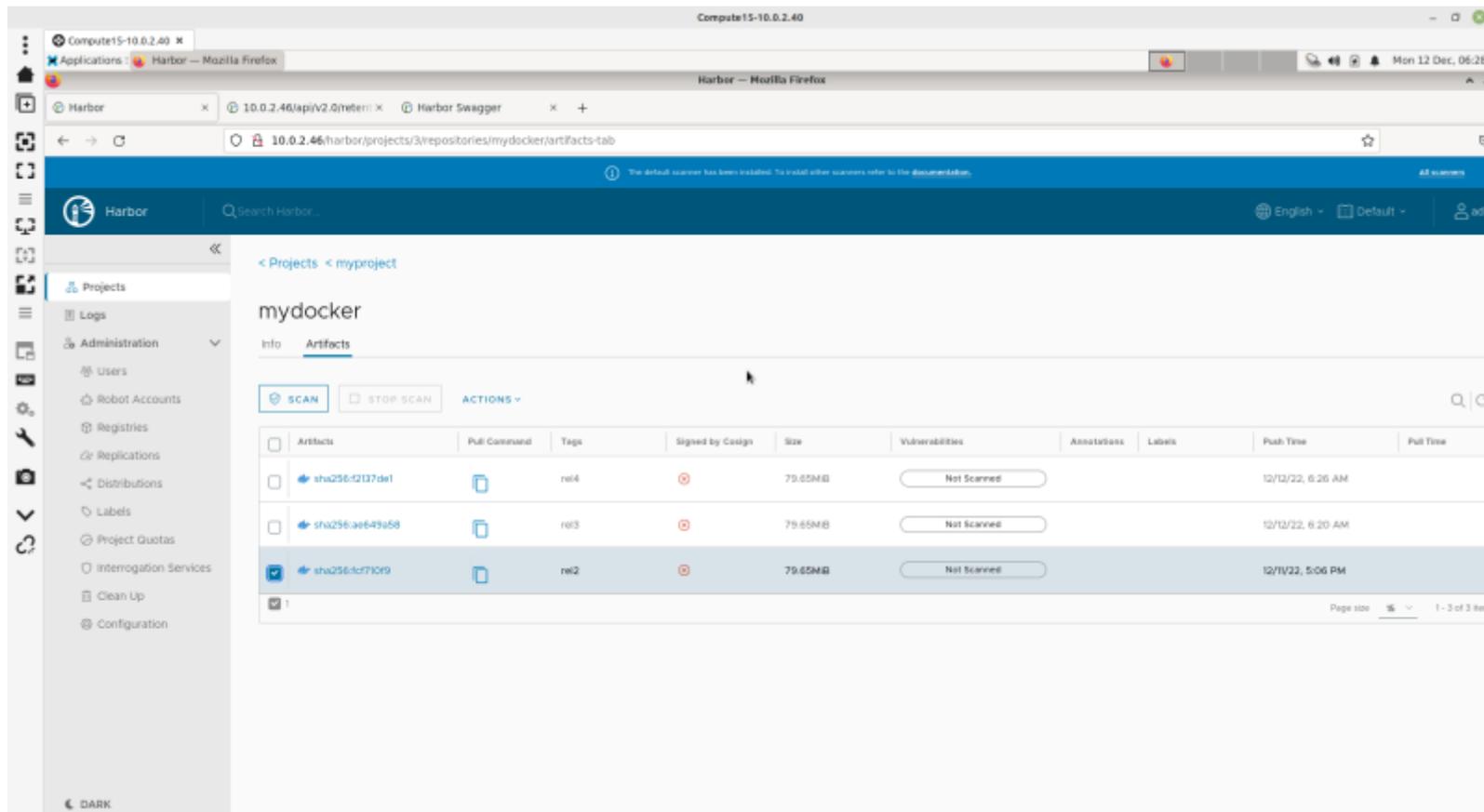
Procédez maintenant au **push** :

```
root@debian11:~/myDocker# docker push 10.0.2.46:80/myproject/mydocker:rel4
The push refers to repository [10.0.2.46:80/myproject/mydocker]
7c7558b01004: Layer already exists
74ddd0ec08fa: Layer already exists
rel4: digest: sha256:ae649a586f5d9dcae71440bf88eb98d95c3cc86d40f27eb54fa68e65869c3b48 size: 736
```



Important : Notez qu'il est possible de faire un push car le tag rel4 n'existait pas.

Dans l'interface de Harbor, cliquez sur **Projects > myproject > myproject/mydocker** et cochez **rel2** :

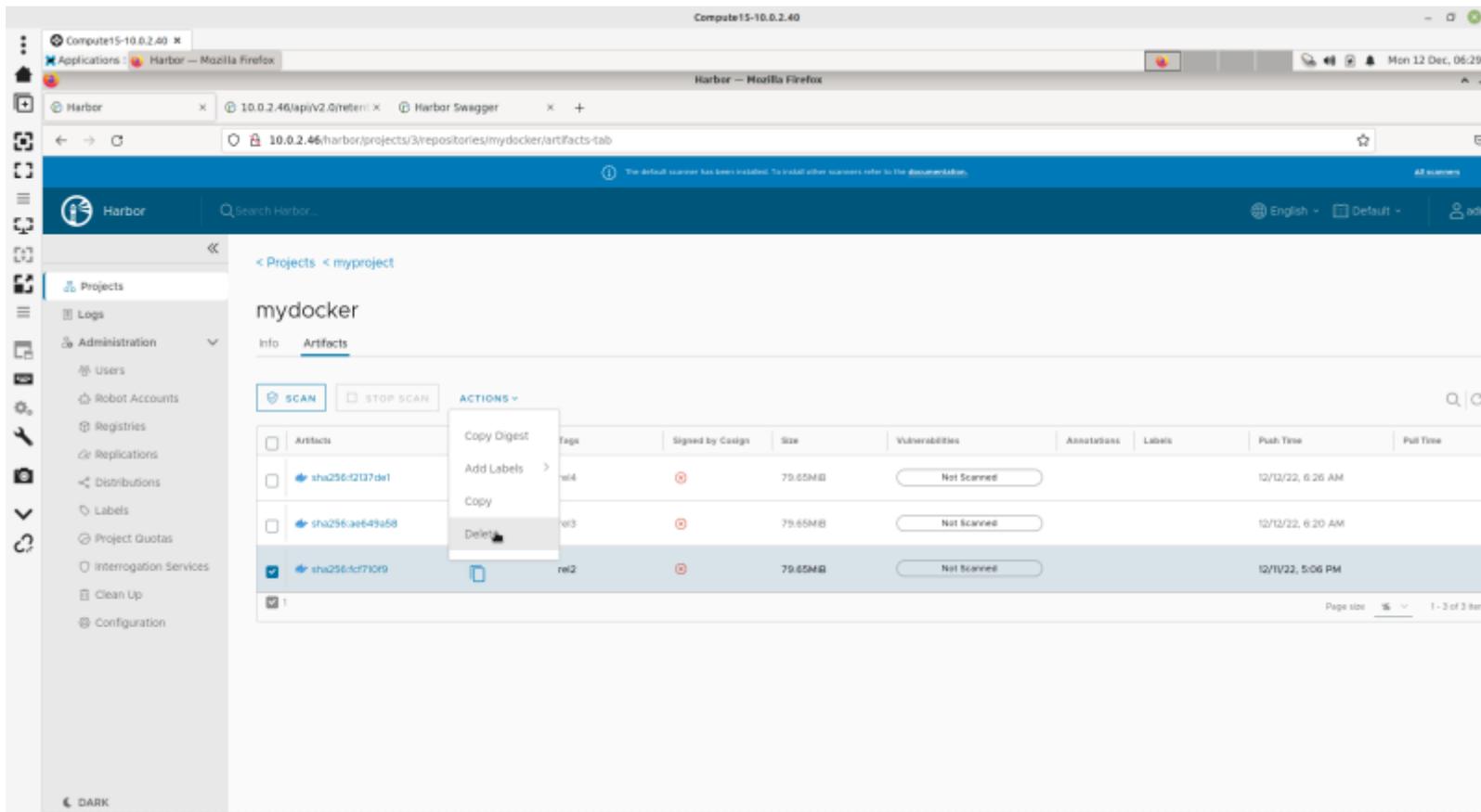


The screenshot shows the Harbor web interface in a browser window. The URL is `10.0.2.46/harbor/projects/3/repositories/mydocker/artifacts-tab`. The page title is `< Projects < myproject` and the main heading is `mydocker`. The `Artifacts` tab is active. There are buttons for `SCAN`, `STOP SCAN`, and `ACTIONS`. A table lists the artifacts:

Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
<input type="checkbox"/>	<code>sha256-12137de1</code>	<code>rel4</code>		79.65MB	<code>Not Scanned</code>			12/12/22, 6:26 AM	
<input type="checkbox"/>	<code>sha256:a6649a58</code>	<code>rel3</code>		79.65MB	<code>Not Scanned</code>			12/12/22, 6:20 AM	
<input checked="" type="checkbox"/>	<code>sha256:c7710f9</code>	<code>rel2</code>		79.65MB	<code>Not Scanned</code>			12/14/22, 5:06 PM	

At the bottom right of the table, it says `Page size 10 1 - 3 of 3 items`. The left sidebar contains navigation options like `Projects`, `Logs`, `Administration`, `Users`, `Robot Accounts`, `Registries`, `Replications`, `Distributions`, `Labels`, `Project Quotas`, `Interrogation Services`, `Clean Up`, and `Configuration`. The bottom left corner has a `DARK` theme toggle.

Essayez de supprimer l'Artefact :



Notez l'erreur :

the tag rel2 configured as immutable, cannot be deleted

< Projects < myproject

mydocker

Info Artifacts

SCAN STOP SCAN ACTIONS

<input type="checkbox"/>	Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
<input type="checkbox"/>	sha256:2137de1		rel4		79.65MB	Not Scanned			12/12/22, 6:26 AM	
<input type="checkbox"/>	sha256:a6649a58		rel3		79.65MB	Not Scanned			12/12/22, 6:20 AM	
<input type="checkbox"/>	sha256:c7712f9		rel2		79.65MB	Not Scanned			12/11/22, 5:06 PM	

Page size 1 - 3 of 3 items

Pour pouvoir supprimer l'Artefact, il faut désactiver la règle d'immuabilité :

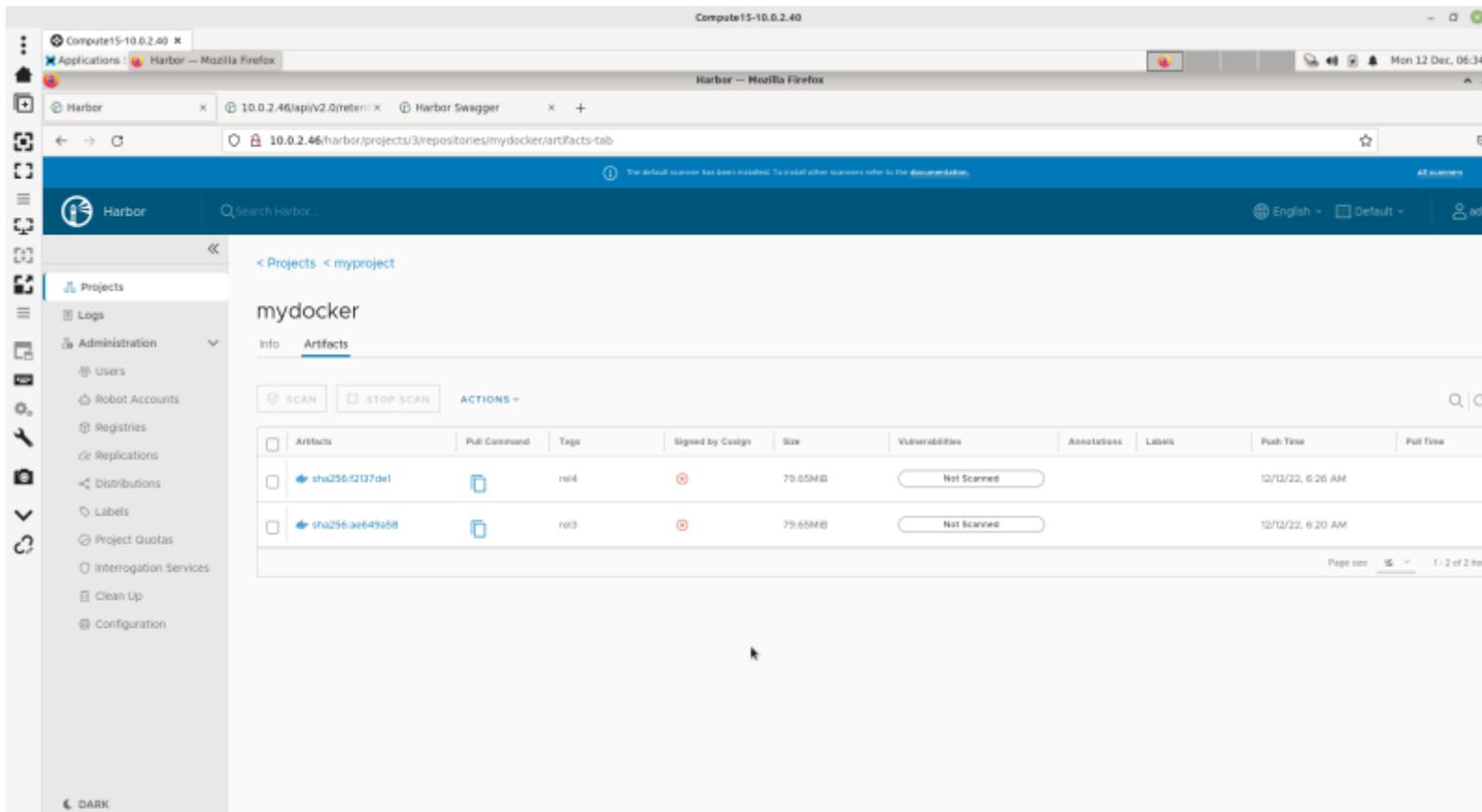
The screenshot shows the Harbor web interface for the 'myproject' configuration page. The 'TAG IMMUTABILITY' tab is selected, and the 'Inmutability rules' section is visible. A dropdown menu is open over the 'ACTION' field, showing options: Deactivate, Edit, and Delete. The 'Deactivate' option is highlighted. The page also shows 'Access Level: Public' and 'Quota used: 82.88MiB of unlimited'.

Maintenant, essayez de supprimer l'Artefact :

The screenshot shows the Harbor web interface in a browser window. The page title is "mydocker" and the sub-page is "Artifacts". A table lists three artifacts, each with a "Not Scanned" status. A context menu is open over the first artifact, showing options: "Copy Digest", "Add Labels", "Copy", and "Delete". The "Delete" option is highlighted, indicating the deletion process is in progress or about to be completed.

Artifacts	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
<input type="checkbox"/> sha256-12137de1	rel4		79.65MB	Not Scanned			12/12/22, 6:26 AM	
<input type="checkbox"/> sha256-ae649a58	rel3		79.65MB	Not Scanned			12/12/22, 6:20 AM	
<input checked="" type="checkbox"/> sha256-1c710f9	rel2		79.65MB	Not Scanned			12/14/22, 5:06 PM	

Vous verrez que la suppression a réussi :



Editez le Dockerfile et changez la valeur de **myVariable** de 8 à 7 :

```
root@debian11:~/myDocker# vi Dockerfile
root@debian11:~/myDocker# cat Dockerfile
FROM centos:latest
MAINTAINER ittraining "infos@ittraining.io"
COPY myEntrypoint.sh /entrypoint.sh
ENV myVariable 7
ENTRYPOINT ["/entrypoint.sh"]
CMD ["mycommand"]
```

Faites un build de **mydocker:rel5** :

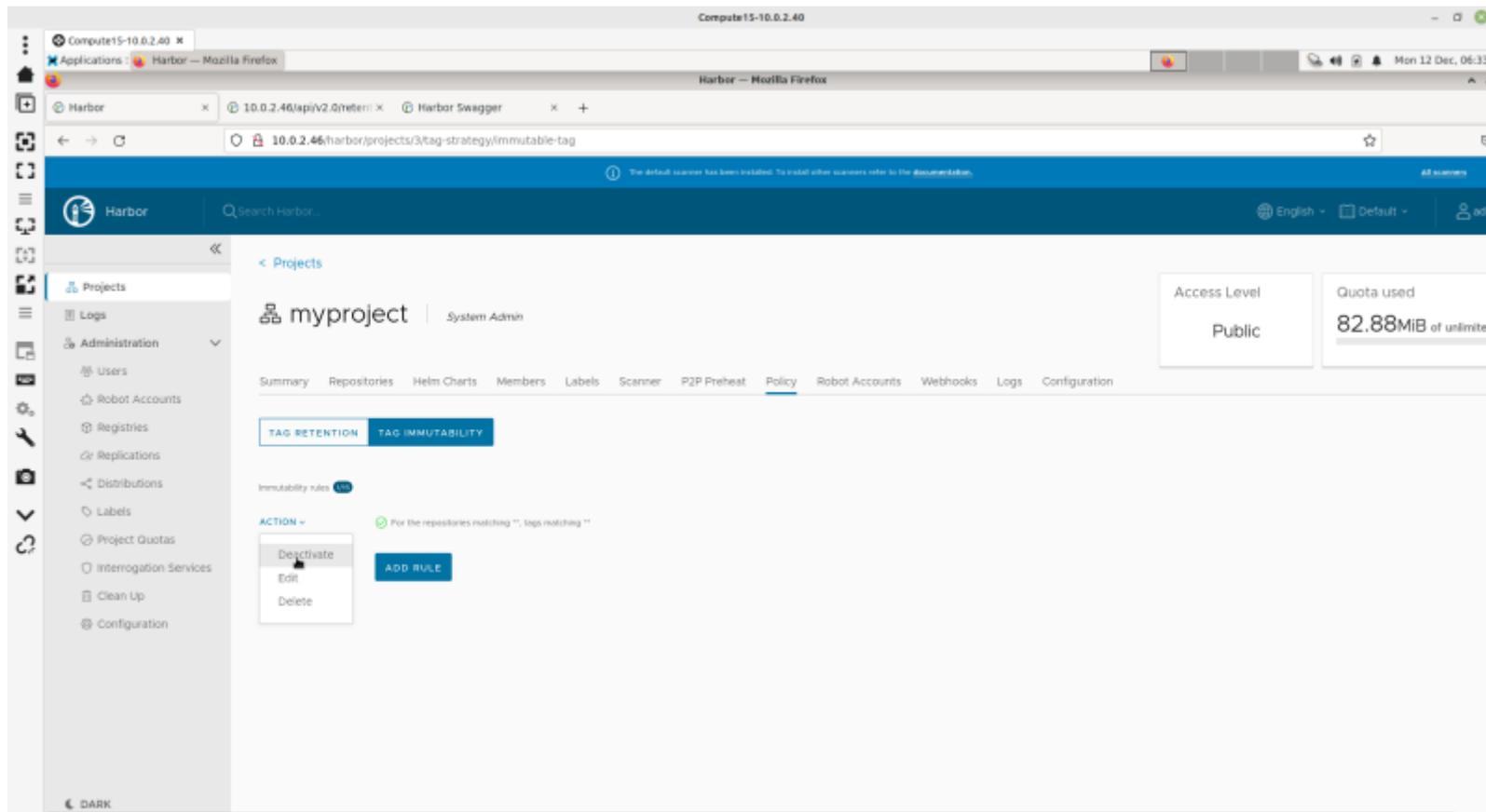
```
root@debian11:~/myDocker# docker build --tag "10.0.2.46:80/myproject/mydocker:rel5" .
Sending build context to Docker daemon 3.072kB
Step 1/6 : FROM centos:latest
----> 5d0da3dc9764
Step 2/6 : MAINTAINER ittraining "infos@ittraining.io"
----> Using cache
----> 7ce8d150d72d
Step 3/6 : COPY myEntrypoint.sh /entrypoint.sh
----> Using cache
----> c1fda8d1e43c
Step 4/6 : ENV myVariable 7
----> Running in b2153c9b0781
Removing intermediate container b2153c9b0781
----> 3b18a7da4503
Step 5/6 : ENTRYPOINT ["/entrypoint.sh"]
----> Running in 81b97d9082ed
Removing intermediate container 81b97d9082ed
----> c283e1e64dca
Step 6/6 : CMD ["mycommand modified"]
----> Running in 7252818184fc
Removing intermediate container 7252818184fc
----> f9466edbef3
Successfully built f9466edbef3
Successfully tagged 10.0.2.46:80/myproject/mydocker:rel5
```

Procédez maintenant au **push** :

```
root@debian11:~/myDocker# docker push 10.0.2.46:80/myproject/mydocker:rel5
The push refers to repository [10.0.2.46:80/myproject/mydocker]
7c7558b01004: Layer already exists
74ddd0ec08fa: Layer already exists
```

```
rel5: digest: sha256:887e7751d8a7271b238f536327c38a867881a9c354ff54d9ab5db45502a160d7 size: 736
```

Pour pouvoir supprimer l'Artefact, il faut désactiver la règle d'immutabilité :



Maintenant, essayez de supprimer l'Artefact :

The screenshot shows the Harbor web interface in a browser window. The page is titled "mydocker" and is under the "Artifacts" tab. A table lists three artifacts, each with a "Not Scanned" status. A context menu is open over the first artifact, showing options: "Copy Digest", "Add Labels", "Copy", and "Delete". The "Delete" option is highlighted, indicating the deletion process is in progress or about to be completed.

Artifacts	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
<input type="checkbox"/>	sha256-12137de1	no	79.65MB	Not Scanned			12/12/22, 6:26 AM	
<input type="checkbox"/>	sha256-ae649e58	no	79.65MB	Not Scanned			12/12/22, 6:20 AM	
<input checked="" type="checkbox"/>	sha256-1c710f9	no	79.65MB	Not Scanned			12/14/22, 5:06 PM	

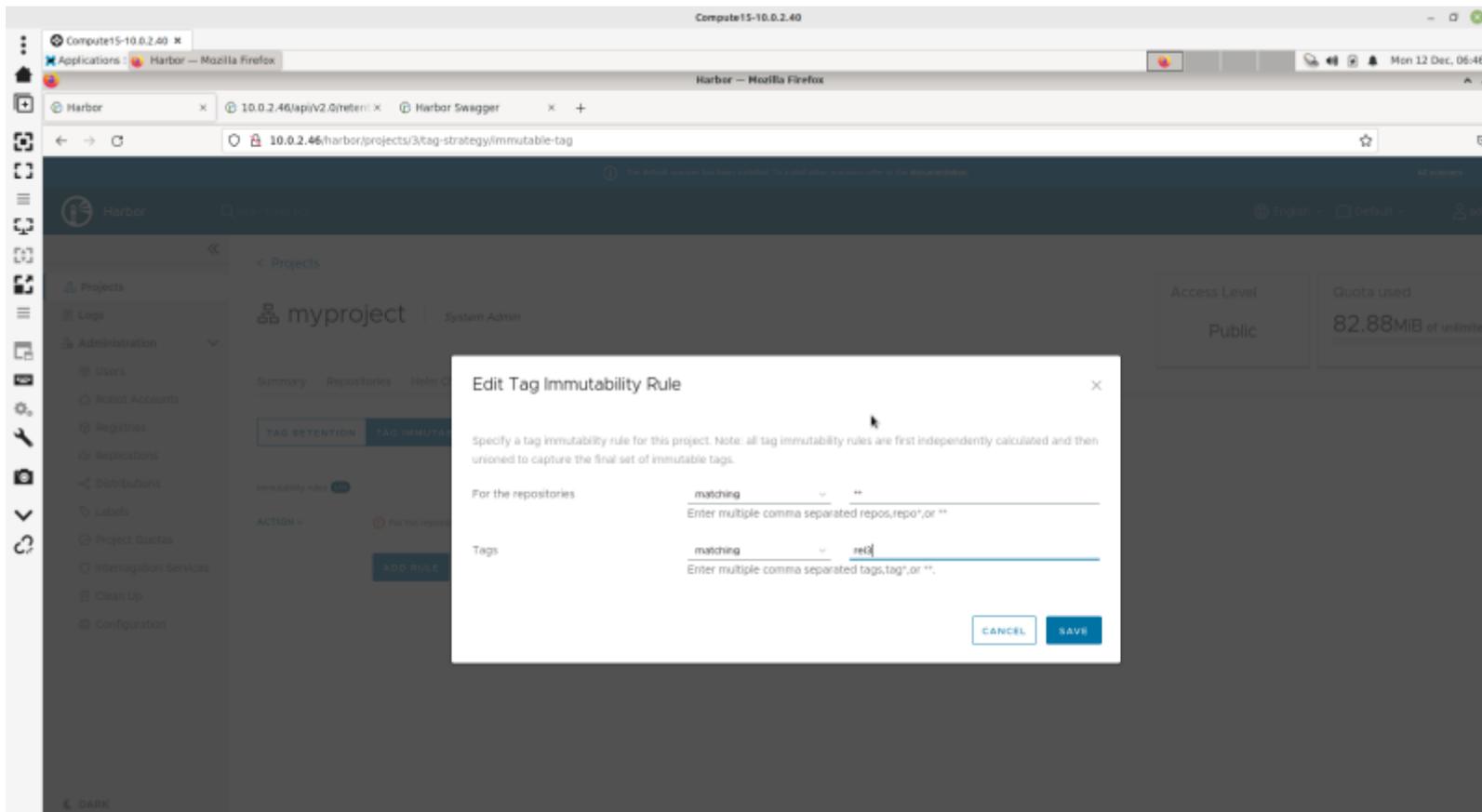
Vous verrez que la suppression a réussi :

The screenshot shows the Harbor web interface for the 'mydocker' project. The 'Artifacts' tab is active, displaying a table of artifacts. The table has the following columns: Artifacts, Pull Command, Tags, Signed by Cosign, Size, Vulnerabilities, Annotations, Labels, Push Time, and Pull Time. Two artifacts are listed:

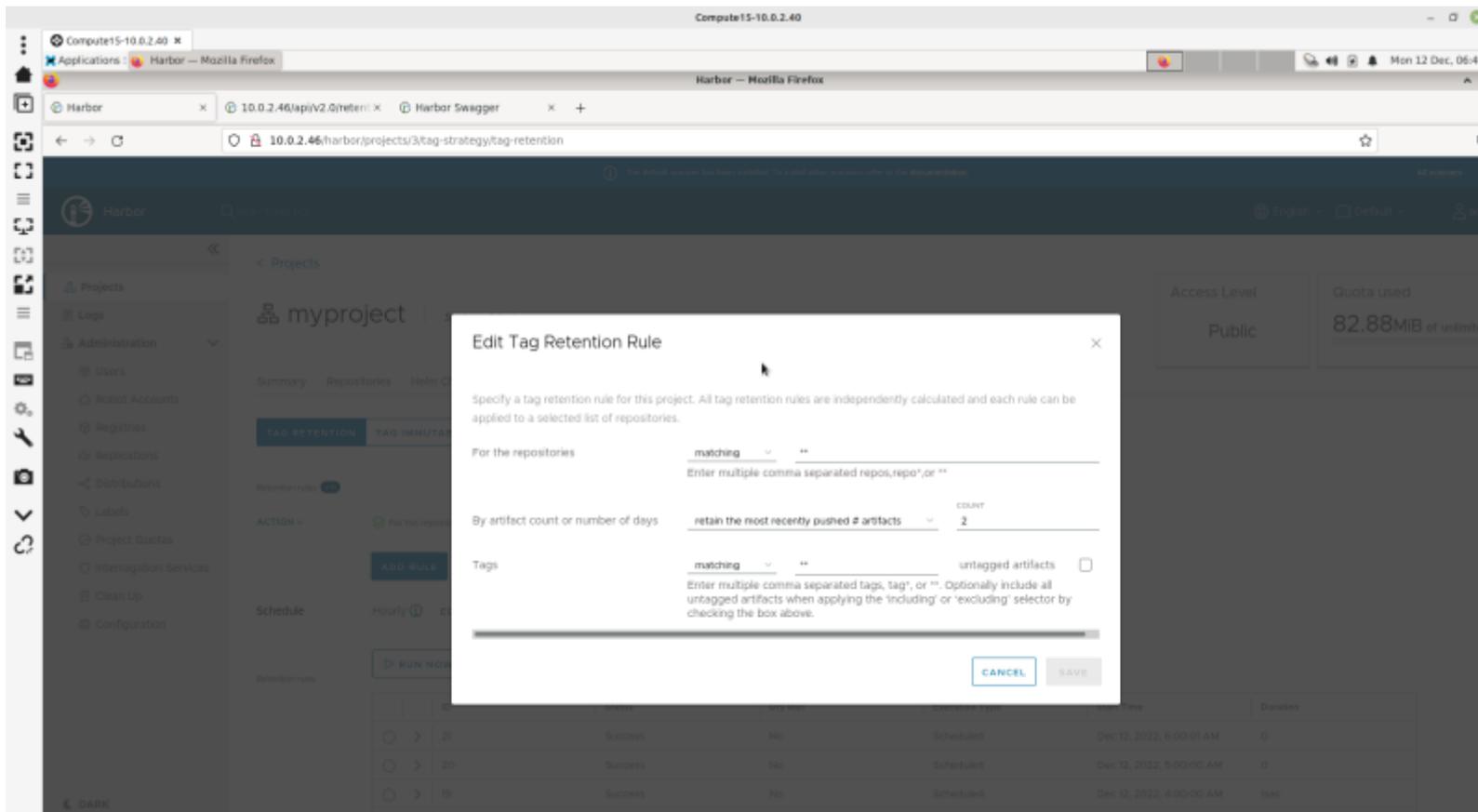
Artifacts	Pull Command	Tags	Signed by Cosign	Size	Vulnerabilities	Annotations	Labels	Push Time	Pull Time
sha256-12137de1		rel4		75.65MB	Not Scanned			12/12/22, 6:26 AM	
sha256-ae649a58		rel3		75.65MB	Not Scanned			12/12/22, 6:20 AM	

The interface also includes a sidebar with navigation options, a top navigation bar with the Harbor logo and search bar, and a footer with a 'DARK' mode toggle.

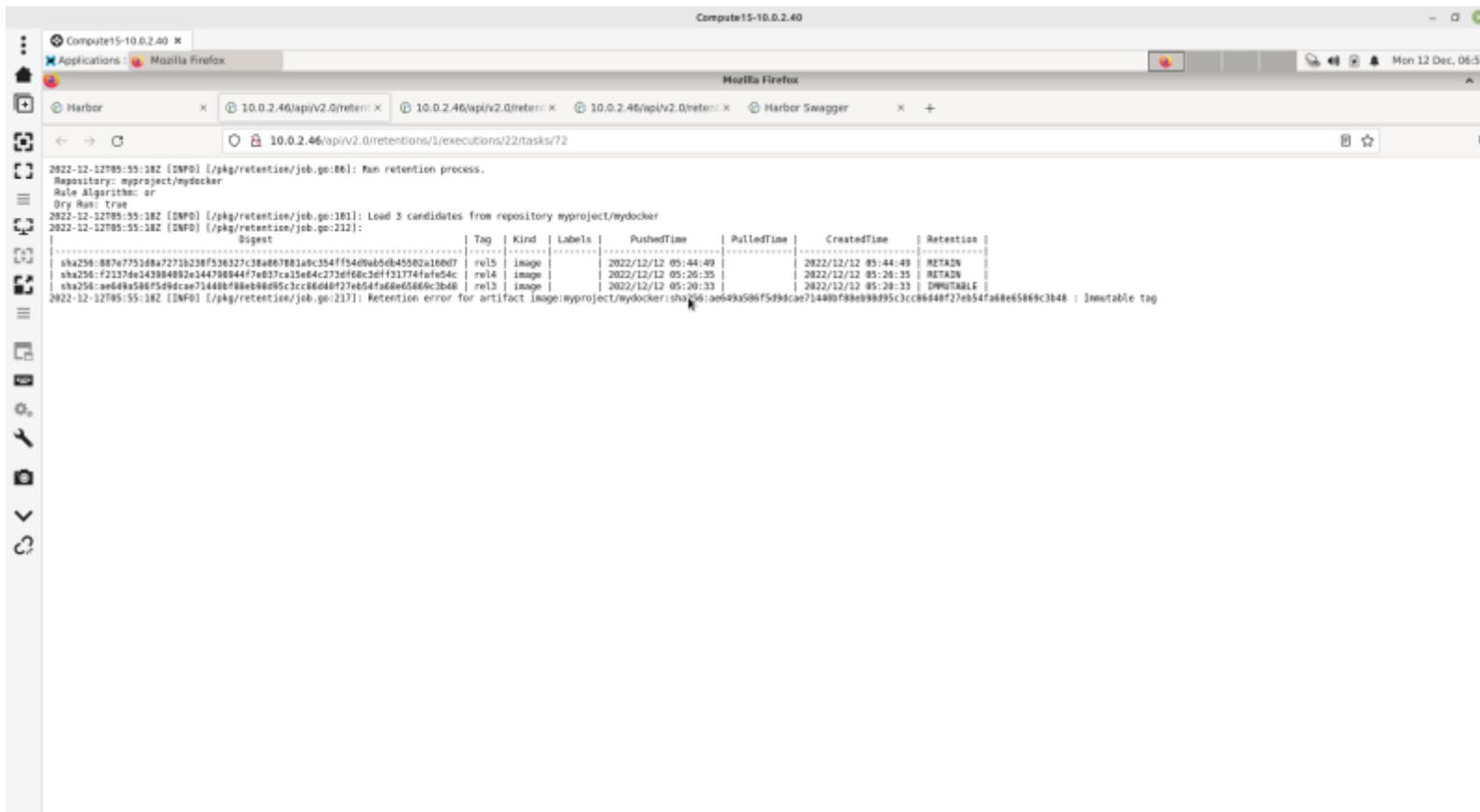
Retournez dans Harbor et modifier la règle d'immutabilité en ne spécifiant que le tag **rel3** puis activez cette règle :



Rappelez-vous que la règle de Rétention de Tags ne garde que les deux Artefacts les plus récents :



Procédez à un Dry Run de la règle de Rétention de Tags et consultez le log de mydocker :



```
2022-12-12T05:55:18Z [INFO] [/pkg/retention/job.go:86]: Run retention process.
Repository: myproject/mydocker
Rule Algorithm: or
Dry Run: true
2022-12-12T05:55:18Z [INFO] [/pkg/retention/job.go:181]: Lead 3 candidates from repository myproject/mydocker
2022-12-12T05:55:18Z [INFO] [/pkg/retention/job.go:212]:
+-----+-----+-----+-----+-----+-----+-----+
| Digest | Tag | Kind | Labels | PushedTime | PulledTime | CreatedTime | Retention |
+-----+-----+-----+-----+-----+-----+-----+
| sha256:887a775188a7273b238f538327c38a867881a6c354ff54d9ab5d845582a168607 | rel5 | image | | 2022/12/12 05:44:49 | | 2022/12/12 05:44:49 | RETAIN |
| sha256:f2137de143984802e344790844f7e037ca15e64c273ef68c3ef121774fefe54c | rel4 | image | | 2022/12/12 05:26:35 | | 2022/12/12 05:26:35 | RETAIN |
| sha256:a649a586f5d9dcae714488f88eb98695c3cc86d8f27eb54fa68e5866c3b48 | rel3 | image | | 2022/12/12 05:20:33 | | 2022/12/12 05:20:33 | IMMUTABLE |
+-----+-----+-----+-----+-----+-----+-----+
2022-12-12T05:55:18Z [INFO] [/pkg/retention/job.go:217]: Retention error for artifact image/myproject/mydocker/sha256:a649a586f5d9dcae714488f88eb98695c3cc86d8f27eb54fa68e5866c3b48 : Immutable tag
```



Important : Notez que la règle d'immuabilité empêche la règle de rétention des tags de supprimer l'Artefact **rel3**.

