Job9

Job 09

Création et Utilisation d'un "registry" local. Et ajouter une « UI » pour le gérer depuis une interface web.

Doc suivie

- https://www.youtube.com/watch?v=If7L4pTH5HA
- https://github.com/kubernetesway/DevOps/blob/main/docker-compose.yaml

1. Création d'un répertoire local_registry_and_ui
2. On crée un fichier docker-compose avec :

https://github.com/kubernetesway/DevOps/blob/main/docker-compose.yaml

```
 GNU nano 7.2                                  docker-compose.yaml
version: '3'

services:
    docker-registry:
        container_name: hub.kubernetesway.in
        image: registry
        ports:
            - 443:443
        environment:
          - REGISTRY_HTTP_ADDR=0.0.0.0:443
          - REGISTRY_HTTP_TLS_CERTIFICATE=/certs/server.crt
          - REGISTRY_HTTP_TLS_KEY=/certs/server.key
        restart: always
        volumes:
            - ./volume:/var/lib/registry
            - ./certs:/certs
    docker-registry-ui:
        container_name: docker-registry-ui
        image: konradkleine/docker-registry-frontend:v2
        ports:
            - 8443:443
        environment:
            ENV_DOCKER_REGISTRY_HOST: hub.kubernetesway.in
            ENV_DOCKER_REGISTRY_PORT: 443
            ENV_DOCKER_REGISTRY_USE_SSL: 1
            ENV_USE_SSL: 1
        volumes:
           - ./certs/server.crt:/etc/apache2/server.crt:ro
           - ./certs/server.key:/etc/apache2/server.key:ro
```

3. Dans ce répertoire on crée un répertoire certs
4. Dans ce répertoire certs :

On génère une clé privée (server.key):

openssl genrsa -out server.key 2048

On génère une demande de signature de certificat:

sudo openssl req -new -key server.key -out server.csr \
-subj "/CN=192.168.107.131" \
-addext "subjectAltName = IP:192.168.107.131"

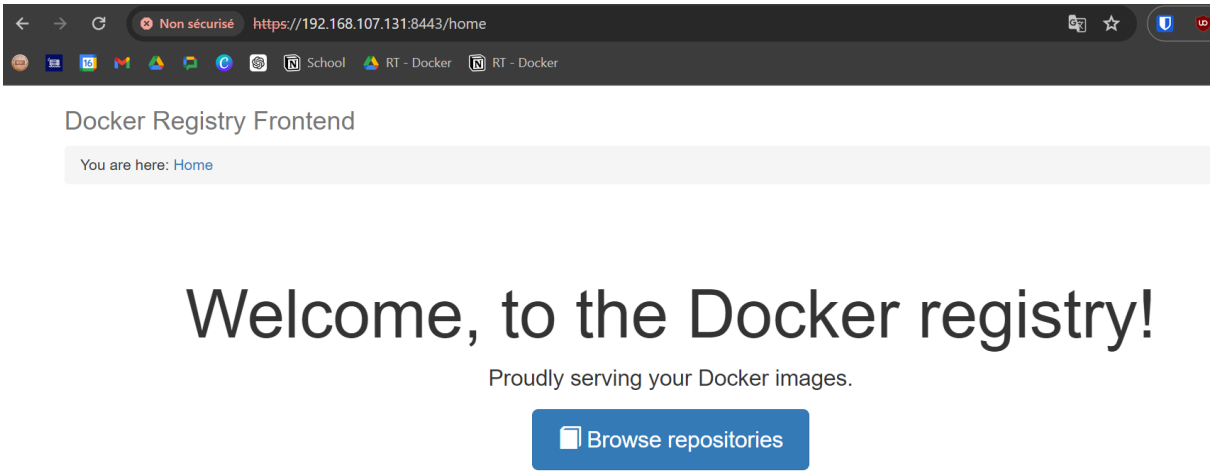
On auto-signe le certificat en utilisant la clé privée

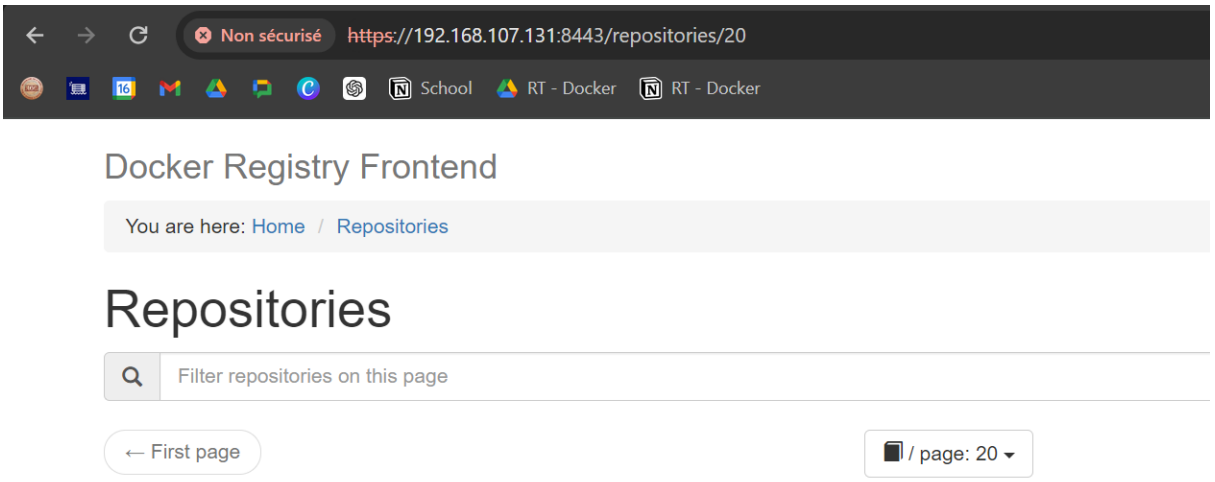
sudo openssl x509 -req -in server.csr -signkey server.key -out server.crt

5. on docker-compose up

```
debian@debian:~/local_registry_and_ui$ ls
certs  docker-compose.yaml  volume
```

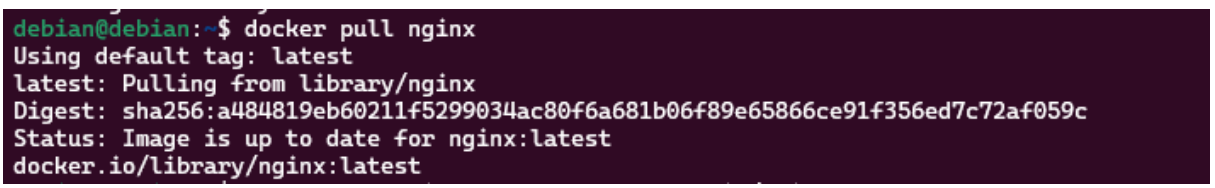
6. on accède à la 192.168.107.131:8443





On y pull une image dans le registry pour voir si ça fonctionne :

docker pull nginx



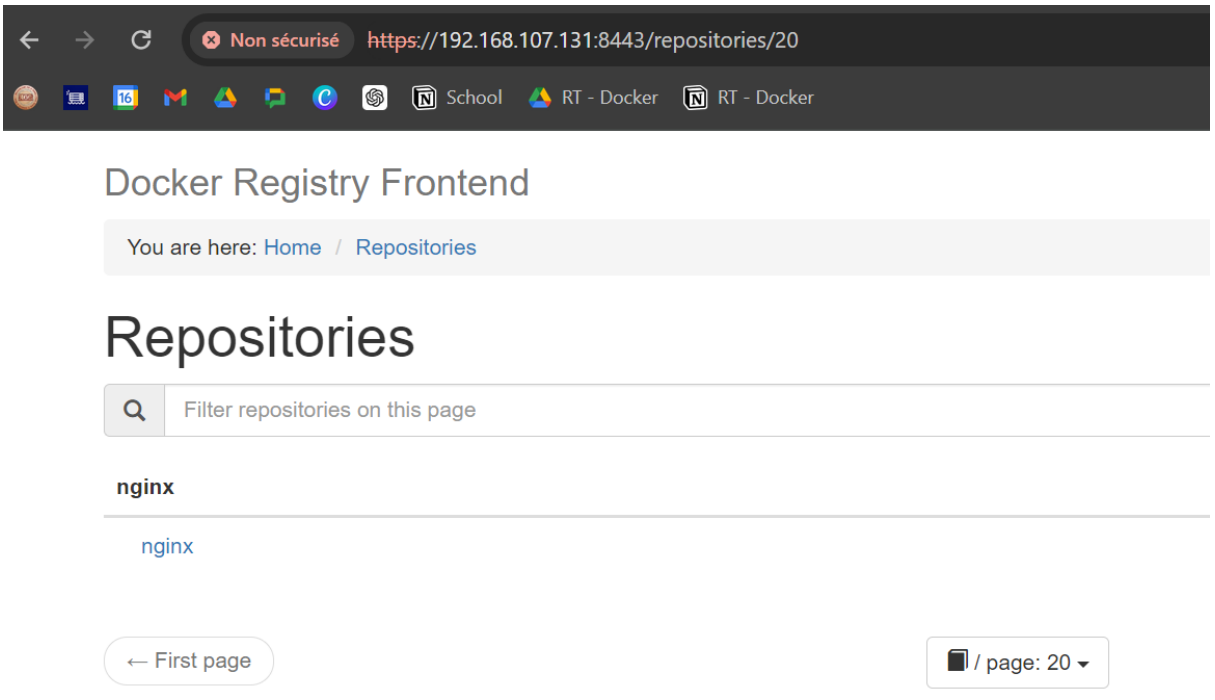
docker tag nginx localhost/nginx

puis

docker push localhost/nginx

```
debian@debian:~$ docker tag nginx localhost/nginx
debian@debian:~$ docker push localhost/nginx
Using default tag: latest
The push refers to repository [localhost/nginx]
14773070094d: Pushed
7d2fd59c368c: Pushed
56f8fe6aedcd: Pushed
9f4d73e635f1: Pushed
747b290aeba8: Pushed
fc1cf9ca5139: Pushed
5d4427064ecc: Pushed
latest: digest: sha256:0e1ac7f12d904a5ce077d1b5c763b5750c7985e524f6083e5eaa7e7313833440 size: 1778
debian@debian:~$
```

Vérifions :



→ ça fonctionne

doc qui aide pour cette erreur

```
debian@debian:~$ docker push 192.168.107.131/nginx
Using default tag: latest
The push refers to repository [192.168.107.131/nginx]
Get "https://192.168.107.131/v2/": x509: cannot validate certificate for 192.168.107.131 because it doesn't contain any IP SANs
```

https://forums.docker.com/t/private-local-registry-tls-certificate-questions/73940

→ Snapshot 6