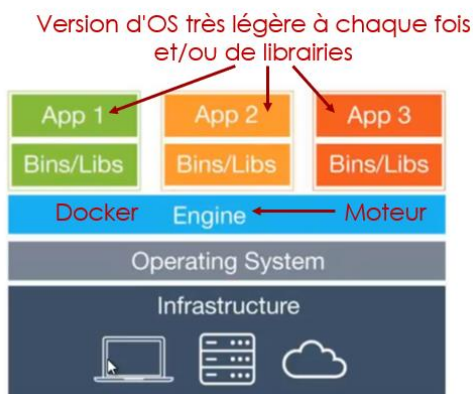
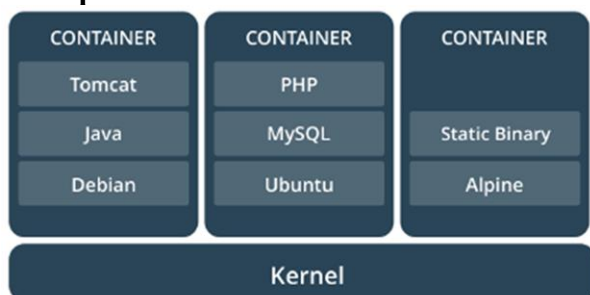


## Docker

### Docker



### Exemples d'utilisation



Les conteneurs Dockers offrent une grande flexibilité qui permet de les créer, déployer, mettre à jour, copier et déplacer d'un environnement à un autre.

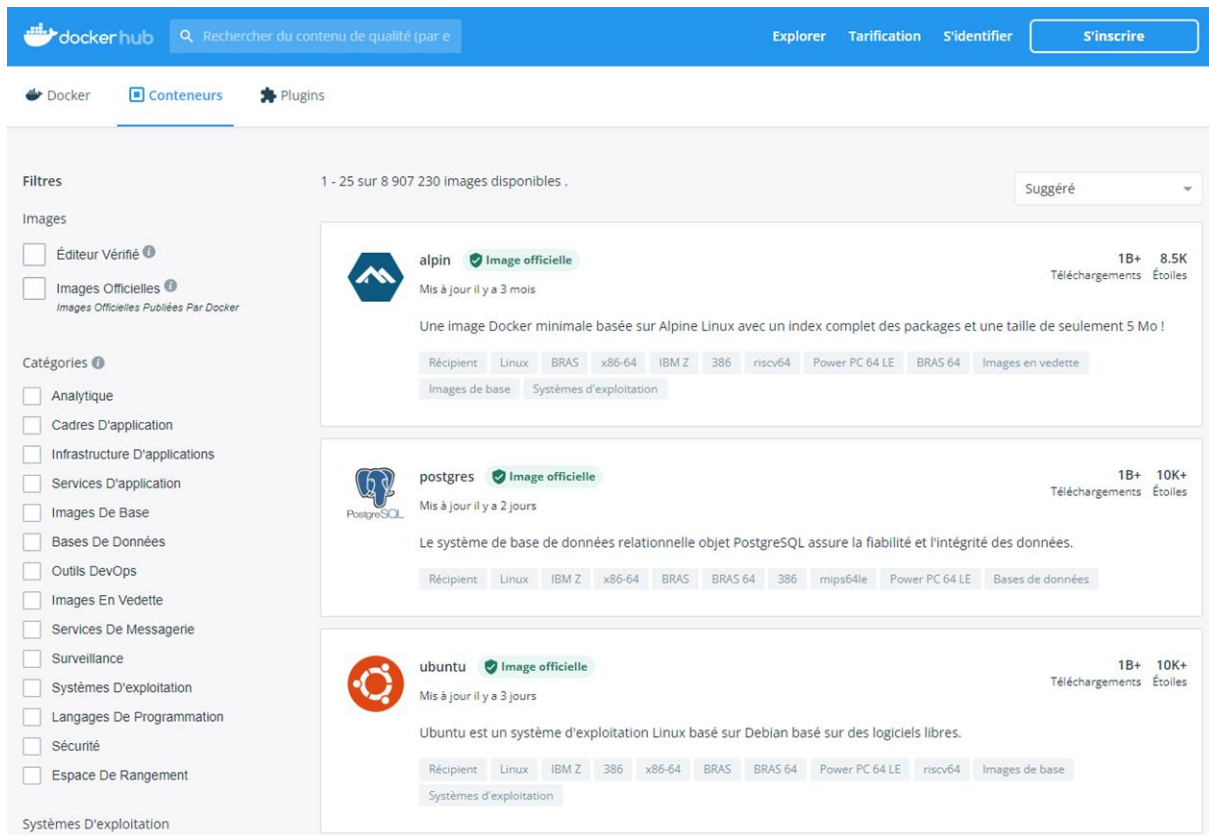
Les conteneurs Docker sont actuellement pris en charge par les principaux acteurs du cloud. Ainsi, un conteneur Docker peut se déployer en de multiples instances sur toutes les principales distributions Linux, Microsoft Windows, et sur toutes les infrastructures.

### Où trouver les images ?

De nombreuses images allant de la simple application comme Nextcloud à un système d'exploitation complet comme Debian sont disponibles sur :

- ☐ Le registre officiel (appelé hub) : <https://registry.hub.docker.com> ;
- ☐ De nombreux dépôts initiés par de « simples » utilisateurs.
- ☐ Il est bien sûr possible de proposer des images, d'en modifier d'autres et de déposer la modification sur le dépôt officiel.

Important lors de son installation désactiver le proxy qui va empêcher via un système de filtrage la recuperation de paquets



## Installation

Avant d'installer Docker, vérifiez qu'une ancienne version ne soit pas déjà présente.

```
apt update
wget https://get.docker.com/
```

## Commandes à tester

```
## List Docker CLI commands
docker
docker container --help

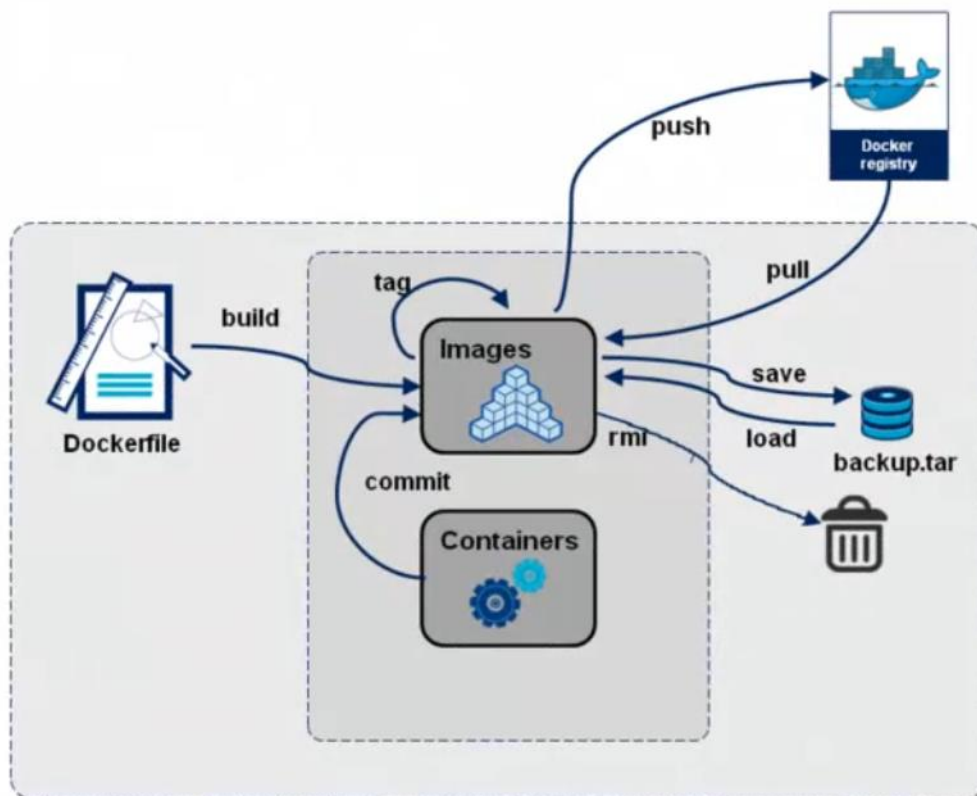
## Display Docker version and info
docker --version
docker version
docker info

## Execute Docker image
docker run hello-world

## List Docker images
docker image ls

## List Docker containers (running, all, all in quiet mode)
docker container ls
docker container ls --all
docker container ls -a -q
```

## Gestion des images



Exectuer la commande run hello-world

docker image -lsdocker

téléchargement de l'image mprasil/dokuwiki

docker pull mprasil/dokuwiki

docker imagesdocker

docker run -d -p 8001:80 --name wiki01 mprasil/dokuwiki

docker exec -it wiki01 bash

docker container ls -a

`docker start NOM_CONTENEUR`

Créer ou récupérer l'image d'une archive

Il faut commencer par compresser l'image au format tar

- `docker save -o <save_image_tar> <image_name>`

Changer de terminal, se connecter en sio et se mettre dans le répertoire *home/sio* (vous ne pourrez pas vous connecter via SSH en root)

- `docker save -o dokuwiki.tar mprasil/dokuwiki`

-image mprasil/dokuwiki                      Putty/WinSEP

- Archive tar => Windows */home/sio*

- Archive tar => nouvelle machine Debian avec Docker

Etape 1

`apt update`

Wget <https://get.docker.com/>

si Erreur :

Solution : désactiver le proxy

`nano /etc/apt/apt.conf`

Docker run hello-world

`bash index.html`

```
root@buster:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:4c5f3db4f8a54eb1e017c385f683a2de6e06f75be442dc32698c9bbe6c861edd
Status: Downloaded newer image for hello-world:latest

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

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

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

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

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

root@buster:~#
```

### **quelque command**

docker = help command

docker container --help = help manage container

### **systemctl status docker**

```

Run 'docker container COMMAND --help' for more information on a command.
root@buster:~# systemctl status docker
• docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-03-11 14:02:31 CET; 11min ago
     Docs: https://docs.docker.com
   Main PID: 3004 (dockerd)
    Tasks: 9
   Memory: 43.1M
   CGroup: /system.slice/docker.service
           └─3004 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

mars 11 14:02:30 buster dockerd[3004]: time="2022-03-11T14:02:30.893404971+01:00" level=warning msg=
mars 11 14:02:30 buster dockerd[3004]: time="2022-03-11T14:02:30.893708552+01:00" level=info msg="Lo
mars 11 14:02:31 buster dockerd[3004]: time="2022-03-11T14:02:31.077077461+01:00" level=info msg="De
mars 11 14:02:31 buster dockerd[3004]: time="2022-03-11T14:02:31.170633800+01:00" level=info msg="Lo
mars 11 14:02:31 buster dockerd[3004]: time="2022-03-11T14:02:31.198629568+01:00" level=info msg="Do
mars 11 14:02:31 buster dockerd[3004]: time="2022-03-11T14:02:31.198960092+01:00" level=info msg="Da
mars 11 14:02:31 buster systemd[1]: Started Docker Application Container Engine.
mars 11 14:02:31 buster dockerd[3004]: time="2022-03-11T14:02:31.237083536+01:00" level=info msg="AP
mars 11 14:06:48 buster dockerd[3004]: time="2022-03-11T14:06:48.725756385+01:00" level=info msg="ig
mars 11 14:08:18 buster dockerd[3004]: time="2022-03-11T14:08:18.562711333+01:00" level=info msg="ig
lines 1-20/20 (END)

```

comment ça marche

## Téléchargement d'une image

docker pull mprasil/dokuwiki

```

root@buster:~# docker pull mprasil/dokuwiki
Using default tag: latest
latest: Pulling from mprasil/dokuwiki
7595c8c21622: Downloading 4.459MB/26.7MB
d13af8ca898f: Download complete
70799171ddba: Download complete
b6c12202c5ef: Retrying in 5 seconds
6297119a6529: Downloading 1.605MB/68.25MB
dba42f81d74f: Waiting
86efb371bce4: Waiting
07eaf366cc68: Waiting
dd02ec152597: Waiting
6dca95daa656: Waiting
97f7h99f9e9e: Waiting

```

docker images « liste des images »

```

root@buster:~# docker images
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
hello-world         latest       feb5d9fea6a5  5 months ago  13.3kB
mprasil/dokuwiki    latest       bb83348cf567  19 months ago  268MB

```

crée un conteneur !

```

root@buster:~# docker run -d -p 8001:80 --name wiki01 mprasil/dokuwiki
c3e575576ef69021984bb880fa2630a6ca11d7bd84650632945f902077347c12

```

-d = lancement en arrière plan

docker exec -it {conteneurID | nom} bash

exemple : docker exec -it wiki01 bash

```

root@buster:~# docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
c3e575576ef6   mprasil/dokuwiki  "/startup.sh run"       5 minutes ago  Exited (0) 54 seconds ago
85d7861351a8   hello-world     "/hello"                33 minutes ago  Exited (0) 33 minutes ago
07372bc61da8   hello-world     "/hello"                35 minutes ago  Exited (0) 35 minutes ago
root@buster:~# _

```

liste des conteneur

reboot d'un  
conteneur

```

root@buster:~# docker exec -it wiki01 bash
Error response from daemon: Container c3e575576ef69021
12 is not running
root@buster:~# docker start wiki01
wiki01
root@buster:~# docker exec -it wiki01 bash
root@c3e575576ef6:/# exit
exit
root@buster:~#

```

Docker start  
NOM\_CONTENEUR

```

root@buster:~# docker run -d -p 8002:80 --name wiki02 mprasil/dokuwiki
15d7fc21923943a3ae9acdf93496380ab6d161dad1646f97c3a67bbf103b50fd
root@buster:~# docker run -d -p 8003:80 --name wiki03 mprasil/dokuwiki
68a22af5efeeae25dac656334e83737f1492f4d9f30e28e1181d1b4172431b1f
root@buster:~#

```

création de 2 autre conteneurs

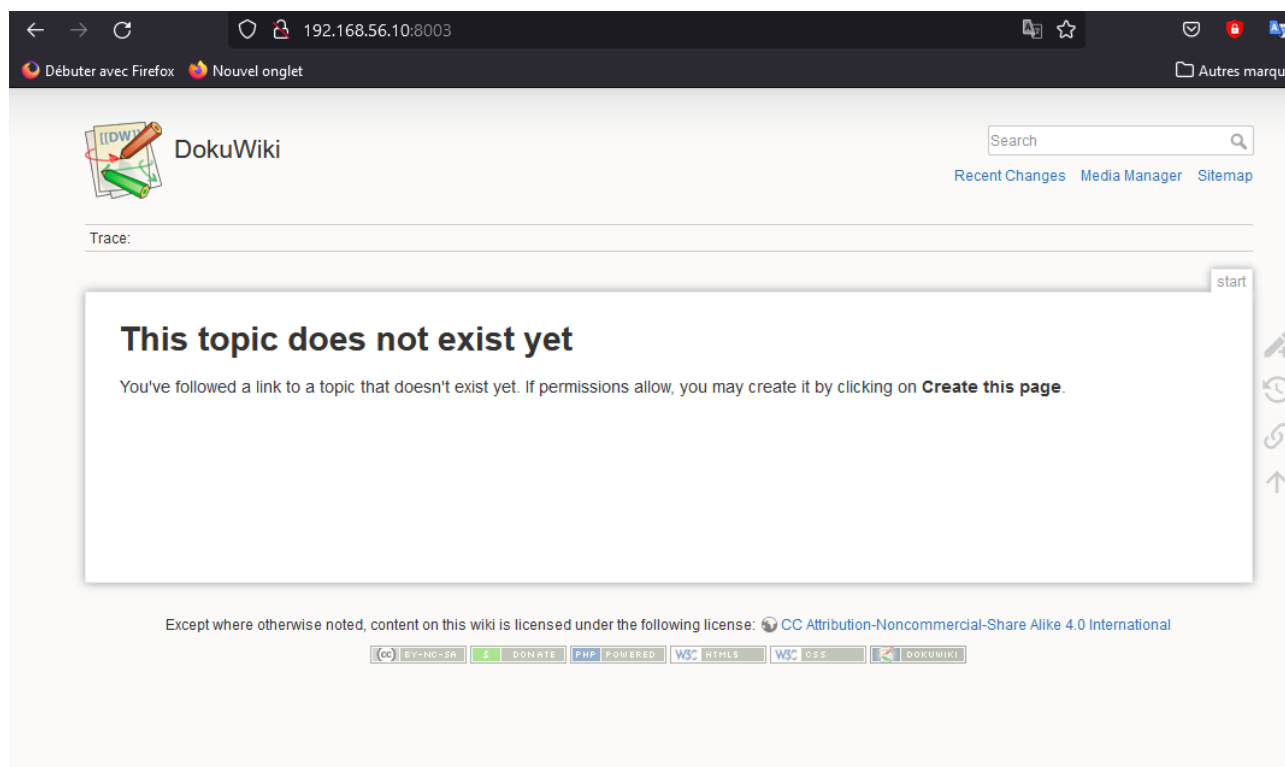
```

CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS
ORTS          NAMES
68a22af5efee   mprasil/dokuwiki    "/startup.sh run"       29 seconds ago Up 28 seconds
.0.0.0:8003->80/tcp, :::8003->80/tcp   wiki03
15d7fc219239   mprasil/dokuwiki    "/startup.sh run"       39 seconds ago Up 37 seconds
.0.0.0:8002->80/tcp, :::8002->80/tcp   wiki02
c3e575576ef6   mprasil/dokuwiki    "/startup.sh run"       12 minutes ago Up 6 minutes
.0.0.0:8001->80/tcp, :::8001->80/tcp   wiki01
85d7861351a8   hello-world         "/hello"                41 minutes ago Exited (0) 41 minutes ago
                                heuristic_blackwell
07372bc61da8   hello-world         "/hello"                42 minutes ago Exited (0) 42 minutes ago
                                dazzling_fermat
root@buster:~#

```

liste des conteneurs

192.168.56.10:8001 /



[Créer ou récupérer l'image d'une archive](#)



en cas de problème de connexion internet pour le téléchargement une sauvegarde locale pour réutilisation est possible

il faut commencer par compresser l'image au format tar

```
root@buster:~# docker save -o dokuwiki.tar mprasil/dokuwiki
```

Changer de terminal se connecter en sio et se mettre dans le répertoire `/home/sio/`

```
mv <image.tar> /home/sio/<image.sio>
```

install ssh

```
root@buster:/home/sio# mv dokuwiki.tar /home/sio/dokuwiki.tar
```

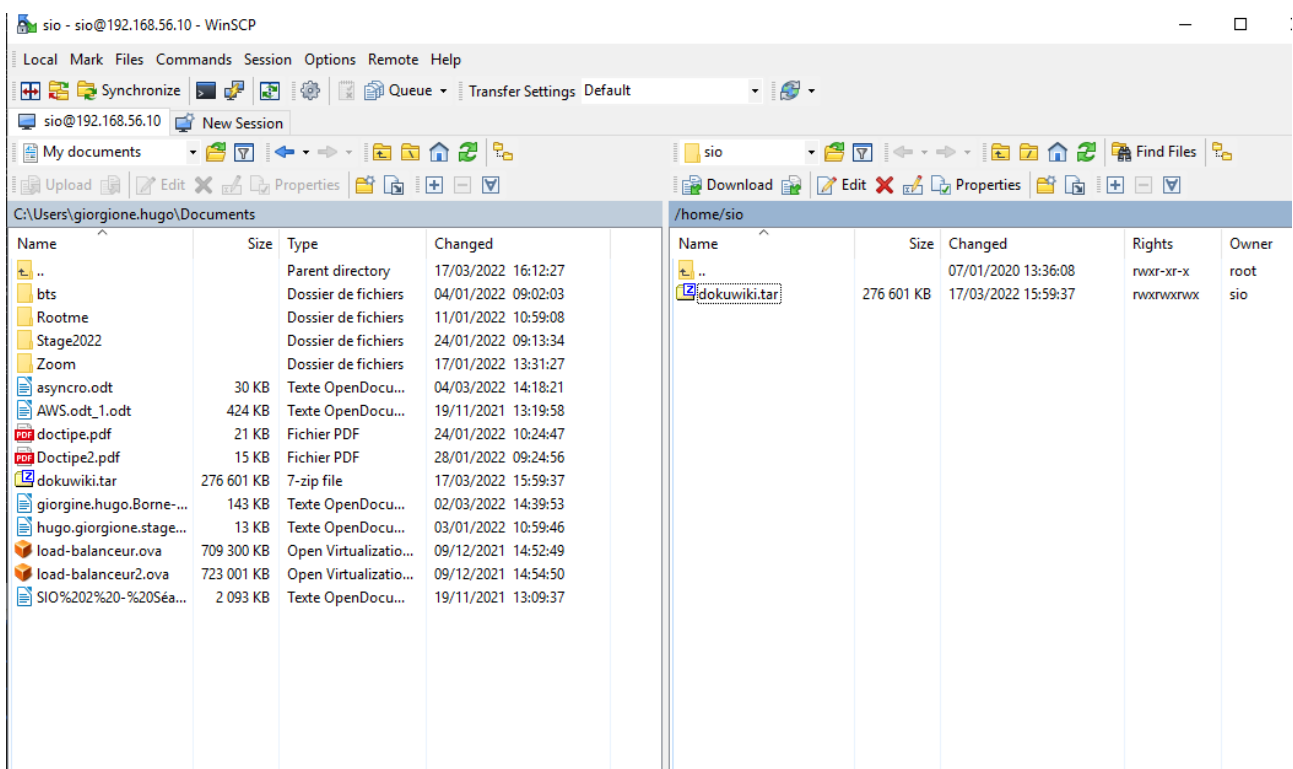
on modifie les droits avec les commandes

```
root@buster:/home/sio# chown sio dokuwiki.tar
```

ou

```
root@buster:/home/sio# chmod 777 dokuwiki.tar
```

ssh sio@192.168.56.10 -p sio



récupération de l'image sur une VM clients

on réinstallé Docker

voir p1

Test de docker avec la commande DOCKER hello-world

verifier si le deamon is running

systemctl status docker,service

ensuite que l'install de base est effectué on va LOAD notre fichier en .tar il faut donc retourner la chercher

apt install ssh

change l'address ip et le réseaux

reboot

WINSCP

on récupéré l'image

on va la déplacer sur le compte root mv dockerwiki.tar /root/dokuwiki.tar

docker load -i dokuwiki.tar

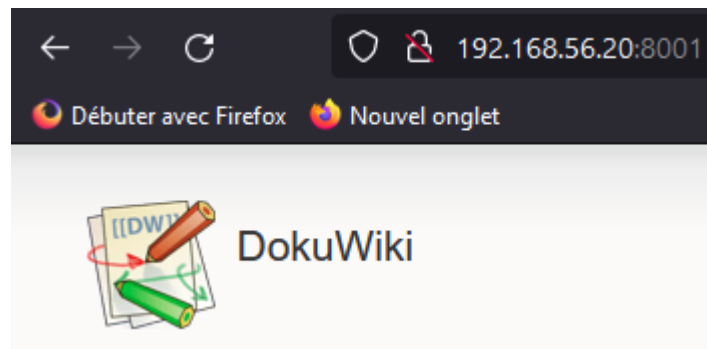
```
root@buster:~# docker load -i dokuwiki.tar
7ef368776582: Loading layer 65.61MB/65.61MB
83f4287e1f04: Loading layer 991.7kB/991.7kB
d3a6da143c91: Loading layer 15.87kB/15.87kB
8682f9a74649: Loading layer 3.072kB/3.072kB
2ae87cb13594: Loading layer 175.8MB/175.8MB
c137ca2ddaed: Loading layer 22.28MB/22.28MB
accc1e5d7eb3: Loading layer 18.46MB/18.46MB
a0e3b14d64d2: Loading layer 4.608kB/4.608kB
8dd2cd7d08e0: Loading layer 4.096kB/4.096kB
ae28848216f4: Loading layer 2.048kB/2.048kB
21692152c9cf: Loading layer 4.608kB/4.608kB
Loaded image: mprasil/dokuwiki:latest
root@buster:~#
```

crée un conteneur !

```
root@buster:~# docker run -d -p 8001:80 --name wiki01 mprasil/dokuwiki  
c3e575576ef69021984bb880fa2630a6ca11d7bd84650632945f902077347c12
```

reboot d'un conteneur

```
root@buster:~# docker exec -it wiki01 bash  
Error response from daemon: Container c3e575576ef69021  
12 is not running  
root@buster:~# docker start wiki01  
wiki01  
root@buster:~# docker exec -it wiki01 bash  
root@c3e575576ef6:/# exit  
exit  
root@buster:~#
```



Etape : Installation NGINX

Objetif : Modifier un conteneur de la page index.html

docker pull nginx

créer un container

rentrer dans le container docker exec -it <nomducontainer> bash

apt update !

Apt install nano

```
nano /usr/share/nginx/html/index.html_
```

ou on va importer depuis la debian directement notre fichier

```
root@buster:~# docker cp index.html nginx:/usr/share/nginx/html/index.html
```

Réalisation d'un script qui va automatiser la modification du fichier dans nginx

```
#!/bin/bash
ajout read -p "indiquer votre prenom " prenom
plate read -p "indiquer votre nom " nom
read -p "indiquer votre classe" classe

eleve="$prenom $nom $classe"

echo " $classe ; $nom ; $prenom "

echo $eleve >> indexbash.html

docker cp indexbash.html nginx:/usr/share/nginx/html/indexbash.html

echo "fin du script"

rm indexbash.html
```

d'un  
tem-  
sur

nginx

téléchargement d'un tample quelconque

Winscp le fichier zip sur a debian

apt install unzip

unzip <fichier>.zip

on déplace les dossier utile a la composition du site web

```
root@buster:~# docker cp index.html nginx:/usr/share/nginx/html/index.html
```

```
root@buster:~# docker cp images nginx:/usr/share/nginx/html/_
```

```
root@buster:~# docker cp assets/ nginx:/usr/share/nginx/html/
```

ensuite on va voir sur notre navigateur

KONTENA

Cluster

Nodes

Workloads

Overview

Pods

Deployments

DaemonSets

StatefulSets

ReplicaSets

Jobs

CronJobs

Configuration

Network

Storage

Namespaces

Events

Access Control

Overview

Pods

Deployments

Pods

30 Items

Name	Namespace	Containers
aws-node-hpz7f	kube-system	
aws-node-r9rgs	kube-system	
calico-node-rwf4g	kube-system	
calico-node-x8slr	kube-system	
calico-typha-78b876fc47-sw5...	kube-system	
calico-typha-horizontal-autos...	kube-system	
cert-manager-789598c8d7-rl4...	kube-system	
coredns-5b7d965bf9-8ttz2	kube-system	
coredns-5b7d965bf9-rpq5q	kube-system	
dashboard-758bd48745-qpsft	kontena-lens	
helm-api-0	kontena-lens	
k8s-resource-applier-6d58457...	kontena-lens	
kube-proxy-svn6b	kube-system	
kube-proxy-v9x5g	kube-system	
kube-state-metrics-76fbc879...	kontena-stats	
license-enforcer-5799c9c94c-...	kontena-lens	
mysql-1568285718-master-0	jakolehlm	
mysql-1568285718-slave-0	jakolehlm	
metrics-server-6bd546f4cd-n...	kube-system	

Pod: calico-node-rwf4g

CPUMemoryNetworkFilesystem

Created

27d 5h 55m ago (2019-08-21T08:50:15Z)

Namespace

kube-system

Status

Running

Node

ip-192-168-81-7.eu-north-1.compute.internal

Pod IP

192.168.81.7

Priority Class

—

QoS Class

Burstable

Labels

controller-revision-hash: 5c9dddc74 k8s-app: calico-node

pod-template-generation: 1

Annotations

scheduler.alpha.kubernetes.io/critical-pod:

Conditions

Initialized Ready ContainersReady PodScheduled

Controlled By

DaemonSet calico-node

Tolerations

9

Show

Affinities

1

Show

Secrets

calico-node-token-8dnkm

Terminal