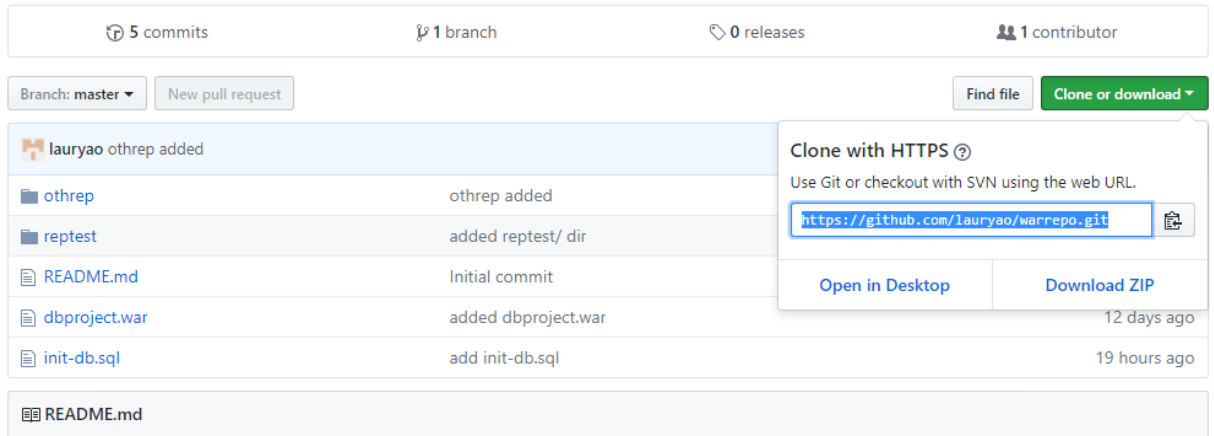


## DOCUMENTATION

- ⇒ Aller dans <https://github.com/lauryao/warrepo> cliquer sur clone or download et copier le lien

*No description, website, or topics provided.*



- ⇒ Aller sur votre toolbox
- ⇒ Taper git clone <https://github.com/lauryao/warrepo.git>
- Nous allons voir si warrepo.war apparait pour cela taper la commande ls

```
$ ls
AppData/
'Application Data'@
Contacts/
Cookies@
Desktop/
Documents/
Downloads/
Favorites/
index.html/
Links/
'Local Settings'@
'Menu Démarrer'@
'Mes documents'@
Modèles@
Music/
myhttp/
myhttp1/
NTUSER.DAT
ntuser.dat.LOG1
ntuser.dat.LOG2
NTUSER.DAT{4e074668-0c1c-11e7-a943-e41d2d718a20}.TM.blf
NTUSER.DAT{4e074668-0c1c-11e7-a943-e41d2d718a20}.TMContainer000000000000000001.regtrans-ms
NTUSER.DAT{4e074668-0c1c-11e7-a943-e41d2d718a20}.TMContainer000000000000000002.regtrans-ms
ntuser.ini
OneDrive/
Pictures/
postgresql/
Recent@
'Saved Games'/
Searches/
SendTo@
testmhttp/
Videos/
'Voisinage d\'impression'@
'Voisinage réseau'@
warrepo/
```

On voit bien le warrepo/

On doit créer deux répertoire

Le 1<sup>er</sup> repertoire comportera le 1<sup>er</sup> dockerfiles et le dbproject.war

On va effectuer la copie du dbprojet.war situé dans warrepo/ dans le 1<sup>er</sup> repertoire.

**Mkdir testmhttp**

**cp warrepo/dbproject.war testmhttp/**

```
Admin_Amertha@DESKTOP-6NR4C4R MINGW64 ~
$ cd testmhttp/

Admin_Amertha@DESKTOP-6NR4C4R MINGW64 ~/testmhttp
$ ls
dbproject.war  Dockerfile
```

- ⇒ On creer l'image de tomcat
- Docker run -d tomcat :8-jre8**
- Vérifier si l'image a été bien créer via la commande docker images

Nous allons créer un Dockerfile via la commande Vi Dockerfile

On utilise l'image de base de tomcat puis on met à jour et on installe le postgresql. On effectue une copie de dbproject.war dans /usr/local/tomcat/webapps/

```
FROM tomcat:8-jre8

RUN apt-get update && apt-get install -y postgresql
COPY ./dbproject.war /usr/local/tomcat/webapps/
```

Attention !! n'oubliez pas de mettre -y car sans le -y l'installation ne sera pas effectuée.

N'oublier pas de sauvegarder via echap : wq!

⇒ On va créer l'image

`docker build -t mytomcat1 :1.1 .`

⇒ verifier si l'image apparait via la commande `docker images`

mytomcat1	1.1	0d1b8335bf8a	About an hour ago	741MB
-----------	-----	--------------	-------------------	-------

⇒ on lance le conteneur :

`docker run -d --name mytomcat1_1 mytomcat1 :1.1`

⇒ on verifie si notre conteneur à bien été lancer via la commande `docker ps`

c8e1cbb7fde9	mytomcat1:1.1	"catalina.sh run"	About an hour ago	Exited (143) About an hour ago	mytomcat1_1
--------------	---------------	-------------------	-------------------	--------------------------------	-------------

Veillez sortir du 1<sup>er</sup> répertoire via la commande `cd ..`

⇒ Créer le 2<sup>ème</sup> repertoire : `mkdir postgresql`

On va effectuer la copie du `init-db.sql` situé dans `warrepo/` dans le 2ème repertoire.

**`cp warrepo/init-db.sql postgresql/`**

aller sur le deuxieme repertoire

**`cd postgresql/`**

créer le 2eme dockerfile

Ce dockerfile a pour but d'effectuer une copie de `init-db.sql` vers `/docker-entrypoint-initdb.d/`

```
FROM postgres:9.5

RUN apt-get update
COPY ./init-db.sql /docker-entrypoint-initdb.d/
```

⇒ On va créer l'image

**`docker build -t mypostgres1:1.1 .`**

⇒ verifier si l'image apparait via la commande `docker images`

mypostgres1	1.1	a0a2c17f8e17	2 hours ago	274MB
-------------	-----	--------------	-------------	-------

⇒ on lance le conteneur :

**`docker run -d --name db mypostgres1:1.1`**

fc2fecaaf48a	mypostgres1:1.1	"docker-entrypoint..."	2 hours ago	Up 2 hours	5432/tcp	db
--------------	-----------------	------------------------	-------------	------------	----------	----

On va lier les deux conteneurs en mappant les ports tout en persistant les volumes

**`docker run -d -P -v dbdata:/var/lib/postgresql/data --name mytomcat1_3 --link db:db mytomcat1:1.1`**

Regarder le port mapper via la commande **`docker ps`**

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e13c4e81f159	mytomcat1:1.1	"catalina.sh run"	About an hour ago	Up About an hour	0.0.0.0:32770->8080/tcp	mytomcat1_3

On peut voir que le port est mappé en 32770

### Lancer l'application

Aller sur un navigateur web taper <http://192.168.99.100:32769/dbproject/accueil.jsp>

On tombe sur la page d'accueil de l'application

## Gestion de stock de chaussures

[Liste des chaussures](#)

[Ajout de chaussure](#)

Aller sur ajout de chaussure

## Ajout de Chaussures au Stock

Genre	<input type="text" value="femme"/>
Marque	<input type="text" value="nike"/>
Modele	<input type="text" value="air III"/>
Couleur	<input type="text" value="bleu"/>
<input type="button" value="VALIDER"/>	

On peut voir la liste de chaussures au stock

### Liste Stock Chaussures

Genre	Modele	Couleur	Marque	Action CMS
femme	air III	blanc	nike	<a href="#">Id</a>
femme	air III	blanc	nike	<a href="#">Id</a>
femme	air III	blanc	nike	<a href="#">Id</a>
femme	air III	bleu	nike	<a href="#">Id</a>