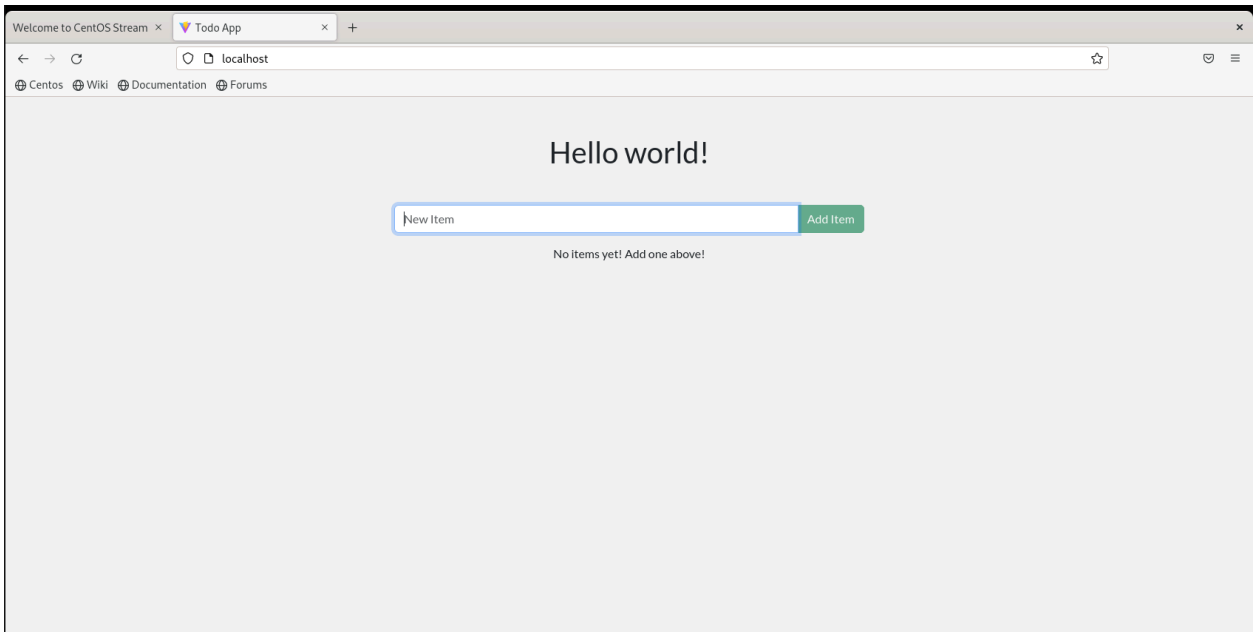
Problem 1:
Composing a docker compose todo app using traefik proxy and react frontend with node for the backend with mysql database and phpmyadmin
First cloning the repo into a todo file
git clone https://github.com/docker/getting-started-todo-app

```
[root@www todo]# ls
backend  client  compose.yml  Dockerfile  LICENSE  README.md
[root@www todo]# docker compose up
[+] Running 38/38
 ✔ mysql Pulled                                                    152.0s
 ✔ proxy Pulled                                                    145.8s
 ✔ phpmyadmin Pulled                                                91.0s
[+] Building 89.7s (23/23) FINISHED                        docker:default
 => [client internal] load build definition from Dockerfile          0.0s
 => => transferring dockerfile: 3.30kB                               0.0s
 => [backend internal] load build definition from Dockerfile         0.1s
 => => transferring dockerfile: 3.30kB                               0.0s
 => [backend internal] load metadata for docker.io/library/node:20   2.7s
 => [client auth] library/node:pull token for registry-1.docker.io   0.0s
 => [backend internal] load .dockerignore                            0.0s
 => => transferring context: 52B                                     0.0s
 => [client internal] load .dockerignore                             0.0s
 => => transferring context: 52B                                     0.0s
 => [backend base 1/2] FROM docker.io/library/node:20@sha256:968ca0550ac  20.8s
```

Running the todo app to check its running



Problem 2:
First creating the volume called my_vol

```
[root@www todo]# docker volume ls
DRIVER    VOLUME NAME
local     my_vol
[root@www todo]#
```

Then attaching it to my_nginx container

```
nvalid argument "-dit" for "--mount" flag: invalid field '-dit' must be a key=value pair
ee 'docker container run --help'.
root@www todo]# docker container run -d -it --mount type=volume,src=my_vol,target=/usr/share/nginx/html  --name my_nginx nginx
d011d4a7bc98d8c65ec7bc7fa28240958a46e4d6f1bb426393587f707ce07e0
```

Then editing theindex.html file inside the container

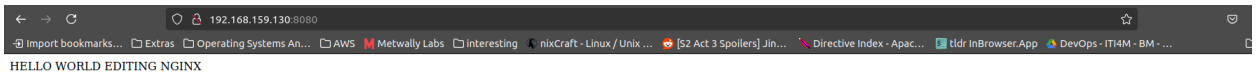
```
[root@www todo]# docker container exec -it my_nginx /bin/sh
# ls
bin  boot  dev  docker-entrypoint.d  docker-entrypoint.sh  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sb
# cd /usr/share/nginx
# ls
html
# cd htmls
/bin/sh: 4: cd: can't cd to htmls
# cd html
# ls
50x.html  index.html
# echo "HELLO WORLD EDITING NGINX" > index.html
# exit
[root@www todo]# ls
backend  client  compose.yml  Dockerfile  LICENSE  README.md
[root@www todo]# docker volume ls
DRIVER    VOLUME NAME
local     my_vol
[root@www todo]#
```

Removing the container

```
[root@www todo]# docker container rm -f my_nginx
my_nginx
[root@www todo]#
```

Creating a new container with port mapping 8080:80

Accessing it persisting the previous edit



Problem 3:

First creating a network with driver of bridge to enable connection between containers

```
[root@www todo]# docker network create --driver=bridge nginx_network
67b772a270e9e7ac48f16fd1bc53aa554f759dea52850509d4a6659b0d0ea486
[root@www todo]# docker network ls
NETWORK ID     NAME            DRIVER    SCOPE
761cd6b92388   bridge          bridge    local
ce8a422d1b71   host            host      local
67b772a270e9   nginx_network   bridge    local
406a01815a13   none            null      local
fb3a58fa0b1d   todo_default    bridge    local
```

Running both containers with the same network

```
[root@www todo]# docker container run -itd -p 4000:80 --network=nginx_network --name nginx_network_1 nginx
700ed7482cbfa382f00df5280bcaae9717e7058bf9e5664765a34f48495cd4bc
[root@www todo]# docker container run -itd --network=nginx_network --name nginx_network_2 nginx
bc023a7c09528ffd2044a622c04e6b9356ae5133111b544ed681a52fb86f744c
```

Finding the ip address using inspect of nginx_network_2

```
                "DriverOpts": null,
                "NetworkID": "67b772a270e9e7ac48f16fd1bc53aa554f759dea52850509d4a6659b0d0ea486",
                "EndpointID": "15733d8e6d5f742378d640f9dea12528d402caf66e4ce7f7d8831576a5e5a1cc",
                "Gateway": "172.19.0.1",
                "IPAddress": "172.19.0.3",
                "IPPrefixLen": 16,
                "IPv6Gateway": "",
                "GlobalIPv6Address": "",
                "GlobalIPv6PrefixLen": 0,
                "DNSNames": [
                    "nginx_network_2",
                    "bc023a7c0952"
                ]
            }
        }
    }
]
[root@www todo]# docker exec -it nginx_network_2 bash
root@bc023a7c0952:/# exit
exit
[root@www todo]# docker exec -it nginx_network_1 bash
root@700ed7482cbf:/# curl 172.19.0.3:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@700ed7482cbf:/#
```

Then finally executing bash into nginx_network_1 and curl on the ip:80