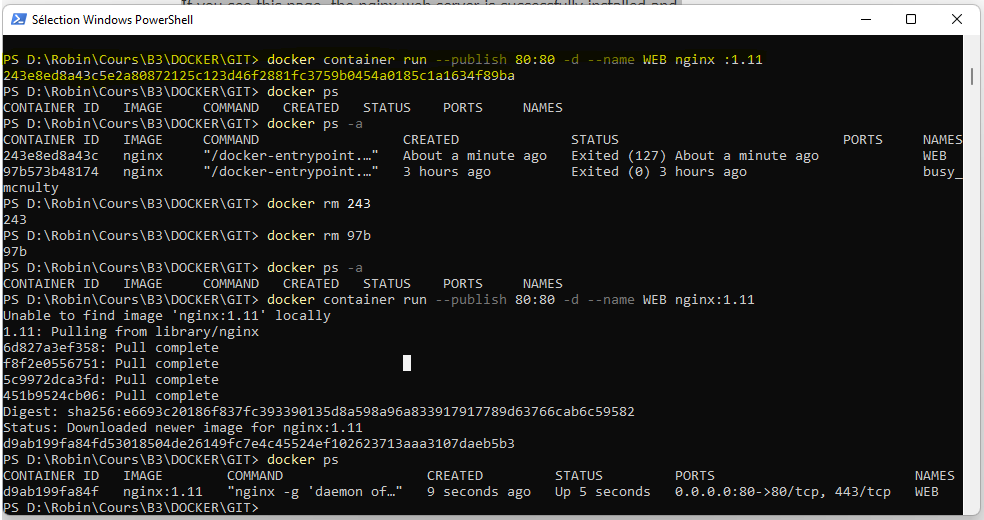
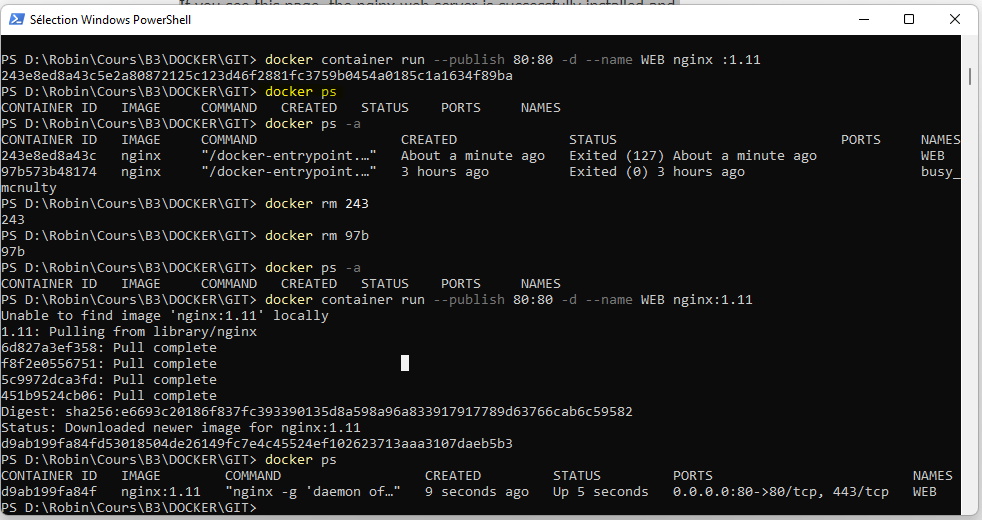
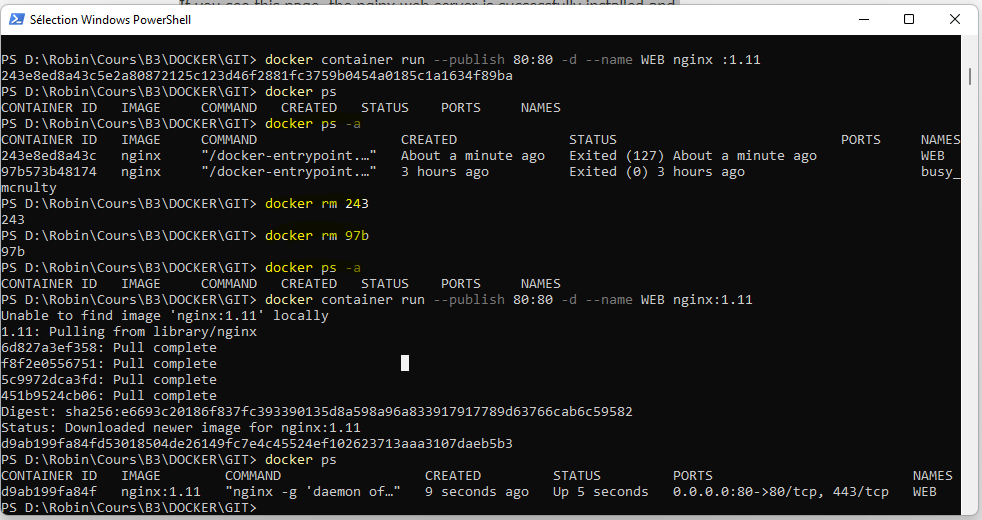
Création de mon container docker container run --publish 80:80 -d --name WEB nginx :1.11



Par la suite je souhaite visualiser mes containers existants. Avec la commande Docker ps

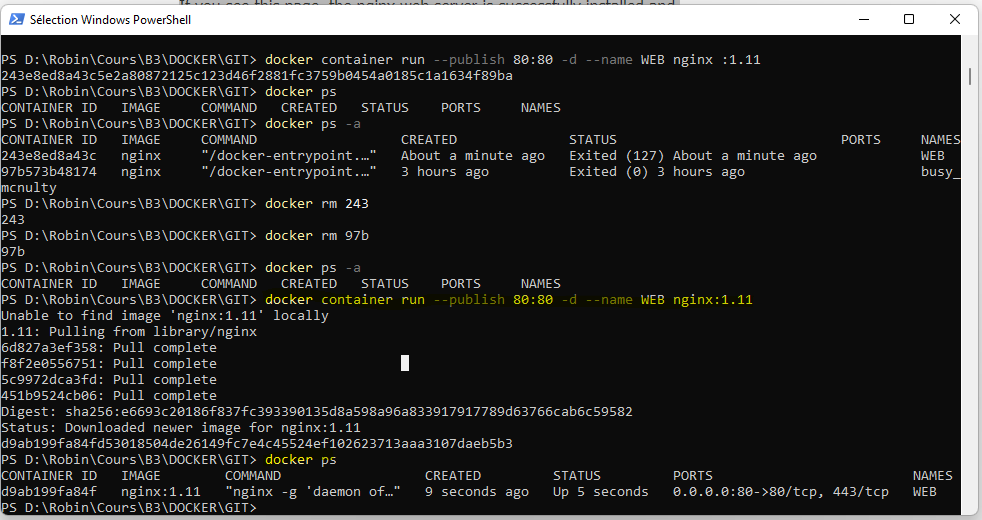


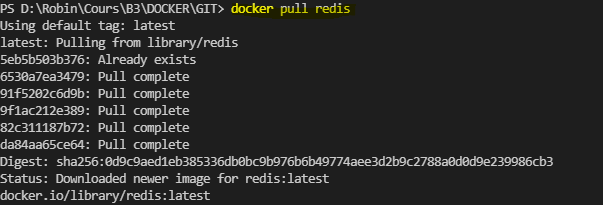
Je n’avais aucun résultat alors j’ai fait un docker ps -a celui-ci m’affiche bien deux containers un créé au préalable et celui créé à l’instant. Par la suite je les ai donc supprimés. Avec la commande docker rm et les 3 premier caractère du CONTAINER ID.

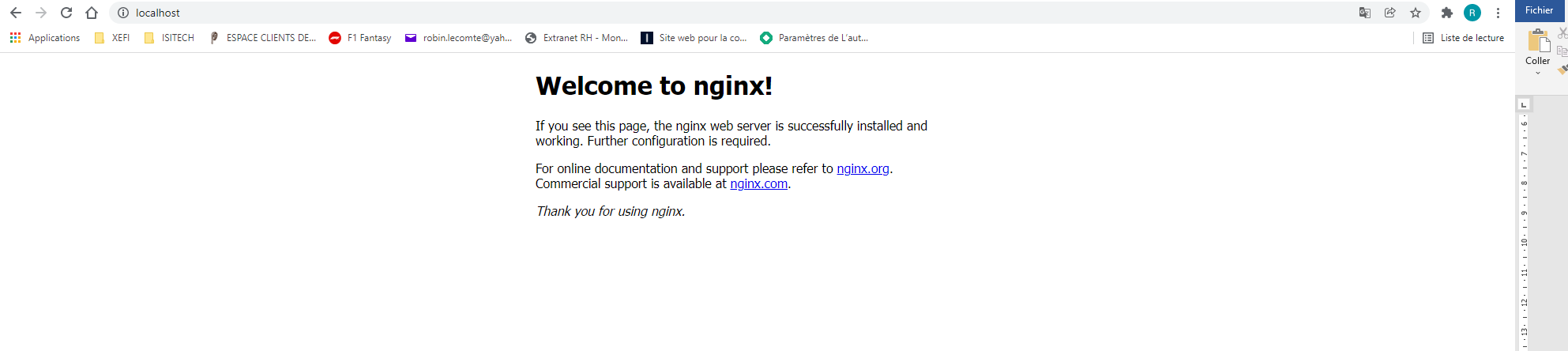


Pour finir, j’ai recréé mon container mais cette foi j’ai modifié la fin de la commande. J’ai collé le nginx :1.11 et mon container c’est créé correctement.

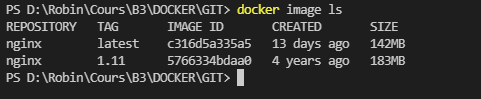
Docker container run --publish 80:80 -d --name WEB nginx:1.11





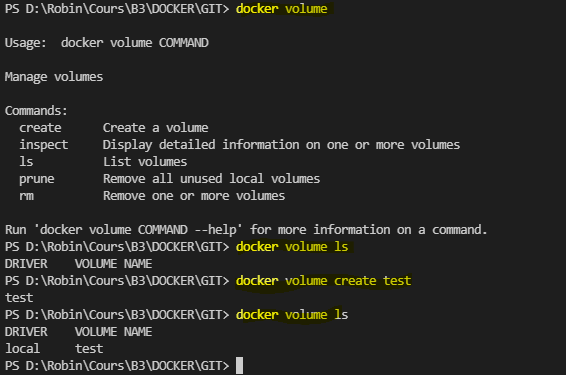
Par la suite, j’ai bien accès via l’adresse web localhost à la pas web suivante.

Par la suite j’ai réalisé la commande docker image ls afin de visualiser les images que j’ai

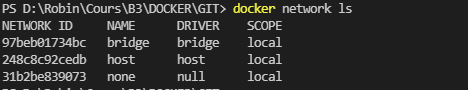


Par la suite, j’ai réalisé la création d’un volume que j’ai nommé test.

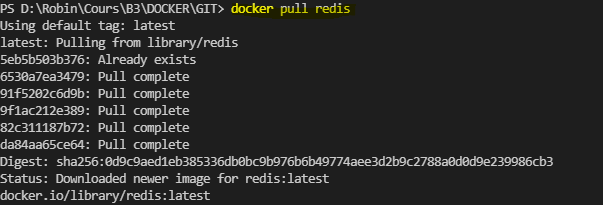
Au début on voit que j’ai aucun volume et par la suite j’en créé un.



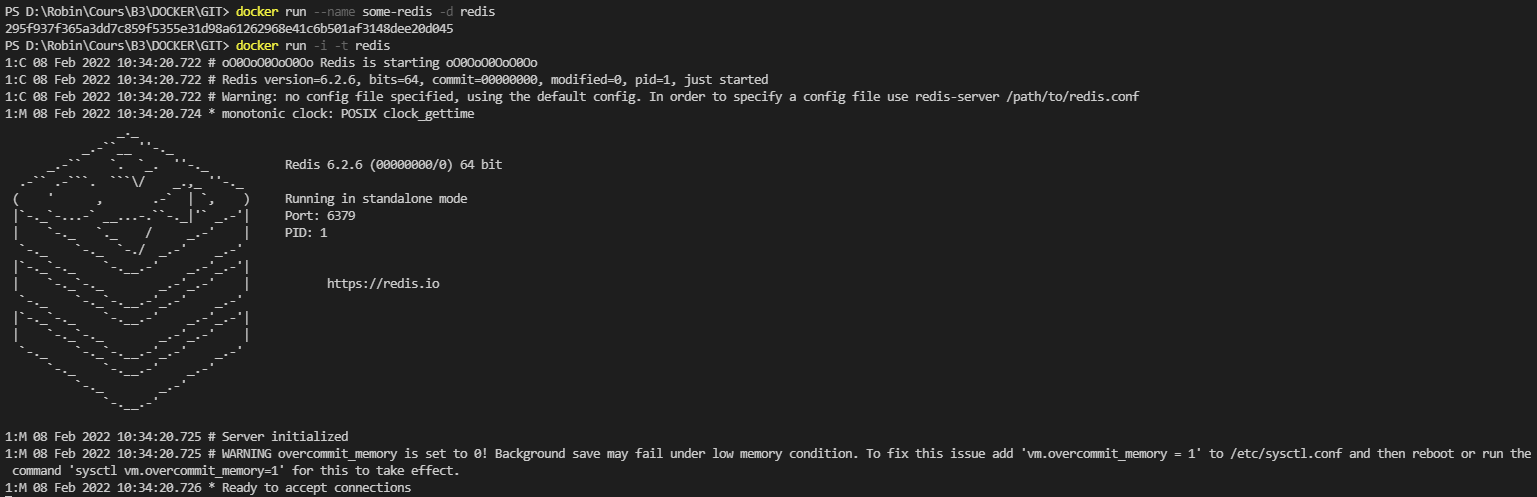
J’ai ensuite réalisé un Docker network afin de visualiser mes différentes cartes réseaux. La host correspond à ma physique sur ma machine, la bridge est une carte réseau virtuel dédié à mon docker, celle-ci me permet d’avoir accès à internet sur mon docker.



Ensuite, j’ai réalisé la commande Docker pull redis afin de télécharger l’image redis.



Ensuite, j’ai démarré mon instance redis puis je suis rentré dedans afin de pouvoir la manager.



Par la suite j’ai créé un dockerfile Dans celui-ci j’ai défini Les différents argument que je souhaitais lui apporter. Ici je souhaite mettre en place l’image d’une debien9. On retrouverra tous les arguments qui ont été commenté sur la capture d’écran. Une foi sauvegardé, j’ai effectué la commande docker build . dans le terminal.

