**Compte rendu PROGWEB :**

**Henky**

Etudiant 1 : Yohann LETELLIER 22317638

Etudiant 2 : Kirsten CHANG 22311567

Année 2023-2024

Master 1 Cybersécurité

Professeur : Monsieur CHARROUX

# 

[**Mini application Henky : serveur et client 2**](#_qwdvyhb4m213)

[**Base de données dans le cloud : MongoDB 2**](#_jkikdzoe6jnn)

[**Copies d’écran des Google Labs 4**](#_12vl6dpc1qb5)

# 

# **Mini application Henky : serveur et client**

Henky est un réseau social qui permet à un utilisateur de créer un compte ou de se connecter via un nom d’utilisateur et un mot de passe. Cet utilisateur peut ensuite poster des Henks, qui sont des messages, dans la plateforme Henky, afin d'interagir avec d’autres utilisateurs. Il peut également rechercher un Henk en particulier en entrant un mot ou un simple caractère dans la barre de recherche. Il a également la possibilité de suivre, ou de se faire suivre par d’autres utilisateurs comme tout réseau social, la liste de ces personnes se situant dans le profil de l’utilisateur.

Une manière classique d’ouvrir Henky est de tout d’abord télécharger tous les modules nécessaires, en tapant “npm install” du côté serveur et du côté client. Ensuite, il faut démarrer le serveur en tapant “npm start” du côté serveur, et enfin visualiser et utiliser l’application web en rentrant la même commande du côté client. On obtient alors cette interface :

# 

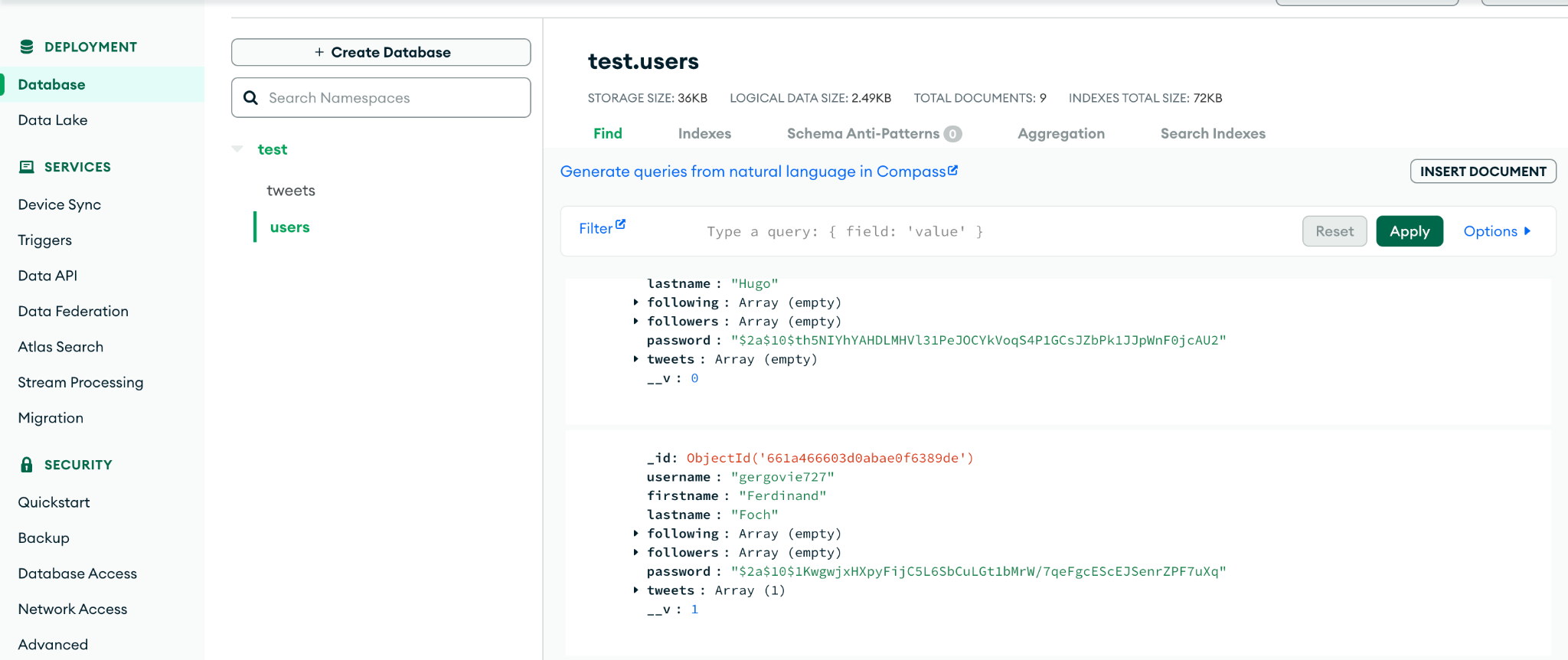
On a donc bien deux services : un serveur et un client qui sont bien reliés entre eux puisque chaque modification sur l’application web est pris en compte dans la base de données dont nous allons parler juste après.

# 

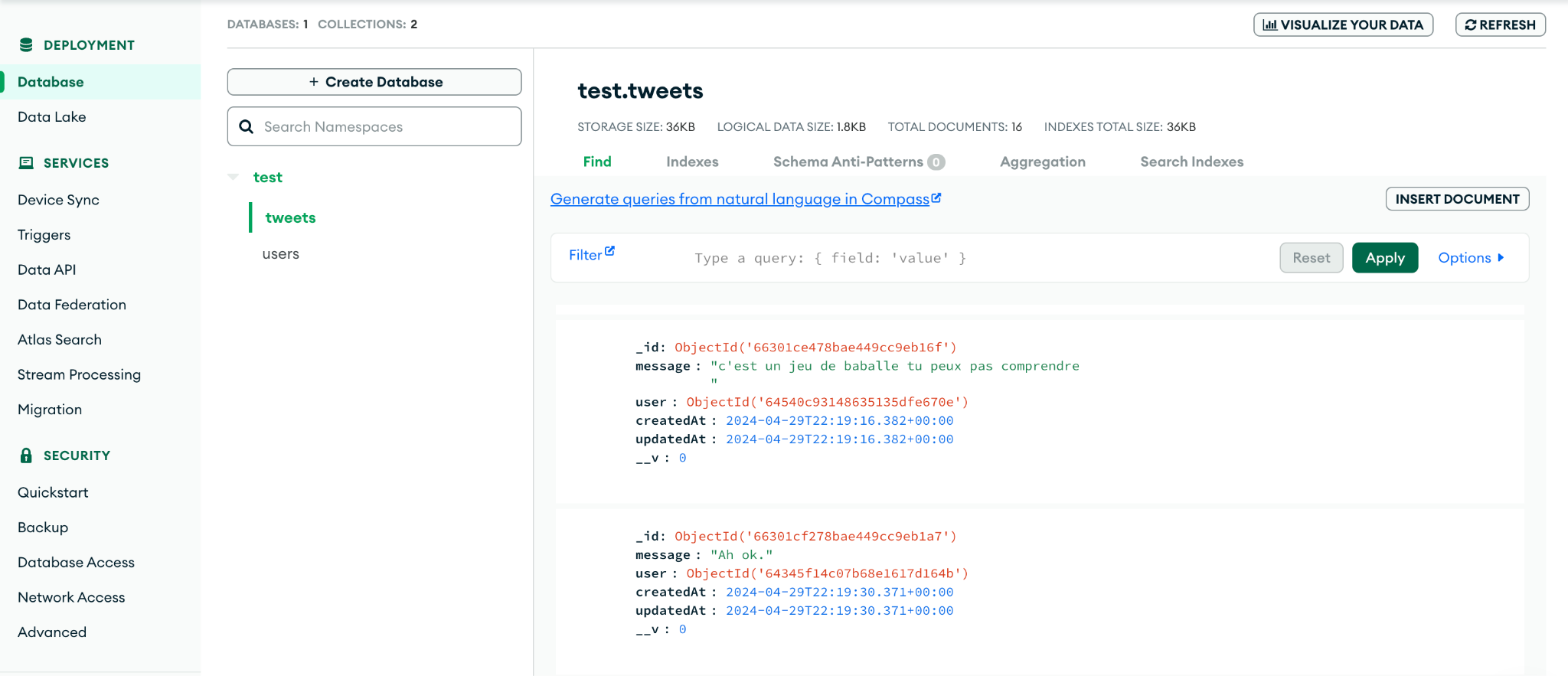
# **Base de données dans le cloud : MongoDB**

Nous avons décidé d’utiliser pour Henky, MongoDB comme base de données NoSQL.

On y trouve tous les utilisateurs, avec leur identifiant unique, leur nom d’utilisateur, leur prénom, leur nom, le nombre de personnes suivies, le nombre de personnes qui suivent cet utilisateur, le mot de passe de leur compte hashé ainsi que leur nombre de Henks (ici tweets) postés.



Le deuxième type de données retrouvé est le Henk (tweet) posté par chaque utilisateur de l’application. On peut y trouver l’identifiant du Henk, le message contenu, l'identifiant de l'utilisateur qui a posté le Henk, la date de création du Henk et enfin la date où il a été modifié pour la dernière fois.

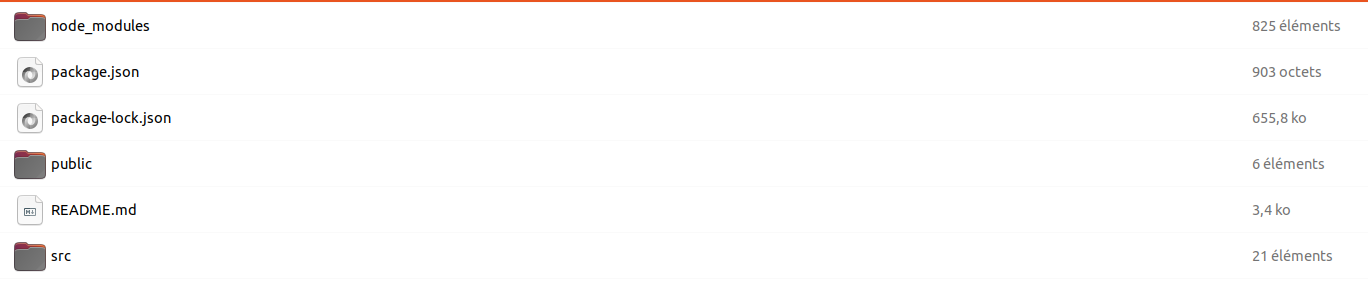


# **Création d’une image Docker et publication dans le DocherHub**

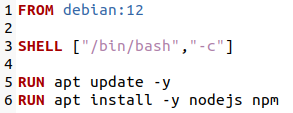
L’application étant répartie en deux, le front-end et le back-end, il faut build deux images dockers, une pour chaque partie.

Les images docker devront contenir tout ce qui permet de faire tourner l’application.

Plaçons-nous d’abord du côté du front-end. Le dossier le contenant se présente ainsi:



Notre fichier Dockerfile devra contenir les instructions pour:

* Avoir l’environnement et tous les paquets nécessaires 
* Mettre tous les fichiers et dossiers nécessaires dans l’image
* Installer les dépendances via NPM et exposer le bon port pour communiquer
* Lancer l’application au démarrage du conteneur 

Nous faisons la même chose avec le back-end.

Ensuite, chacune de ces images est build. On le fait en se plaçant dans les répertoires avec les commandes suivantes:

* sudo docker build -t henky\_front .
* sudo docker build -t henky\_back .

Nous avons par la suite poussé ces images sur le docker hub. La première étape a été de les tag:

* docker tag henky\_front:latest yohannletel/henky\_front:1
* docker tag henky\_back:latest yohannletel/henky\_back:latest

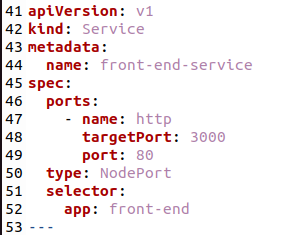
Puis nous les avons push:

* docker push yohannletel/henky\_front:1
* docker push yohannletel/henky\_back:1

# **Création d’un déploiement et d’un service Kubernetes**

Nous avons créé un fichier yaml qui contient les déploiements et services front-end et back-end ainsi que pour la gateway. Ci-dessous l’exemple pour le front-end:

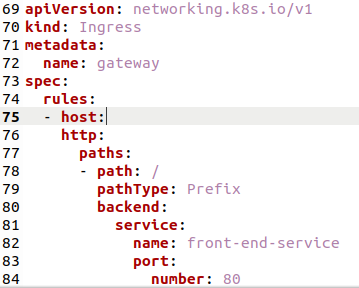




Et voici la gateway:

# **Ajout d’une gateway en local**

Dans le fichier yaml de configuration, on ajoute la gateway:



# **Accès à l’application**

Maintenant que le fichier yaml est bien écrit, on peut l’appliquer.

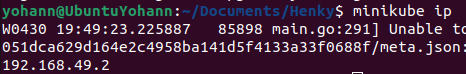
La première chose à faire est de démarrer le cluster Minikube:

“minikube start”

On se place ensuite dans le dossier du fichier, et on l’applique avec la commande suivante:

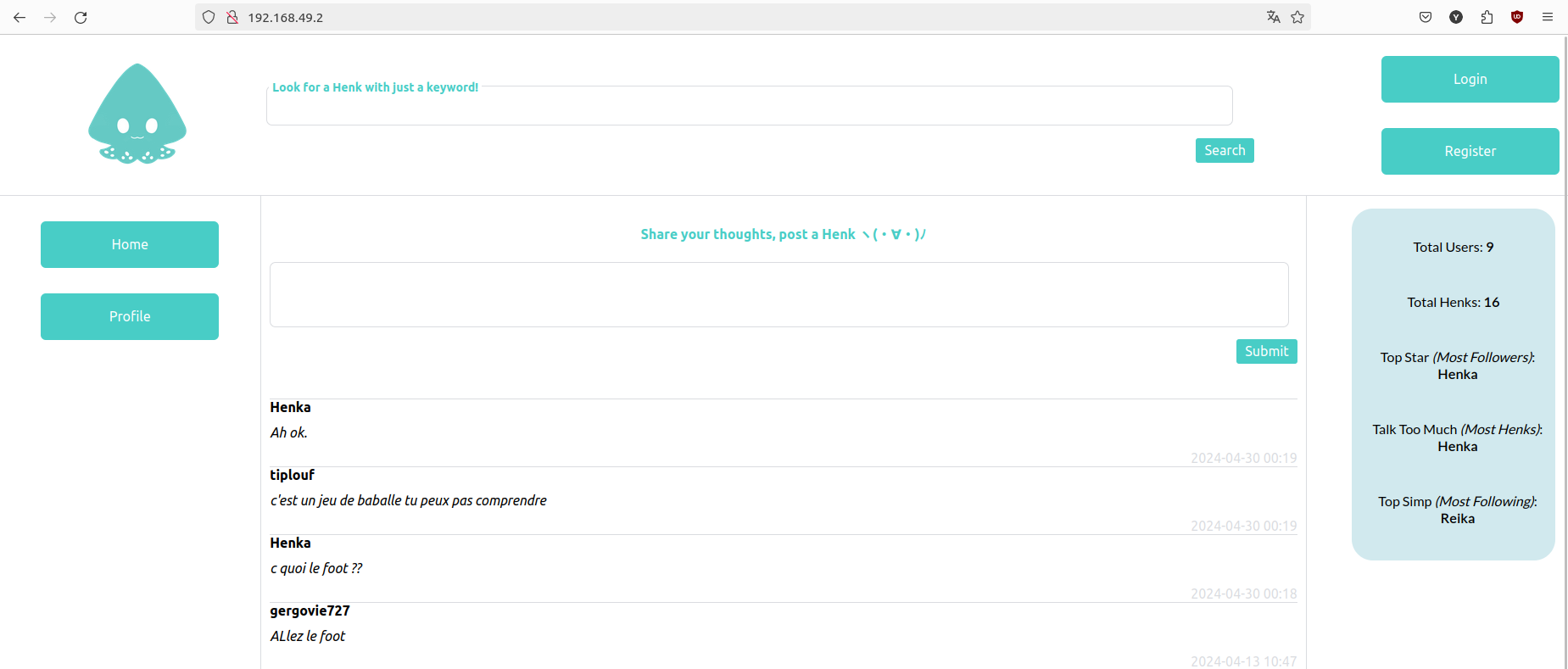
“kubectl apply -f front-back-app.yaml”

Ensuite, on récupère l’IP du cluster minikube avec la commande:



Les messages d’alerte n’ont aucun impact et peuvent être ignorés.

Enfin, on entre cette IP dans la barre d’adresse, et on peut profiter de l’application:



# **Copies d’écran des Google Labs**

