Université Sorbonne Paris Nord
Département d'informatique

Travaux Pratiques

Prise en main de redis en utilisant docker

Denis Linde

27 novembre 2025

Table des matières

Introduction

Ce rapport présente les différentes étapes du travail pratique sur l'utilisation de docker afin de créer un server redis, puis le peupler et le gérer via le client redis.

1 Vidéo redis 1 : installation et prise en main

1.1 Installation et lancement du serveur et client

L'utilisation de docker implique qu'il n'y a pas d'installation à faire, mise à part le téléchargement automatique de la dernière version de redis si celle-ci n'est pas installé au lancement du conteneur.
On lance le conteneur avec la commande :

Listing 1 – bash version

```
~$docker run -d --name redis
```

Puis on lance le serveur avec la commande :

Listing 2 – bash version

```
~$docker exec -it redis bash
```

```
ing@ing-Latitude-5300:~$ docker run -d --name redis redis:latest
5b9162c234aaa53eb1e60c1a1da027f6e8be1a635e8d1fcb2737afc6a402261b
ing@ing-Latitude-5300:~$ docker run -it redis bash
root@f1bdd32912df:/data# ^C
```

FIGURE 1 – Résultats des commandes précédentes

Pour lancer le client c'est encore plus simple : comme le conteneur redis est déjà créé, il suffit de le lancer avec la commande redis-cli en paramètre :

Listing 3 – bash version

```
~$docker exec -it redis redis-cli
```

```
ing@ing-Latitude-5300:~$ docker exec -it redis redis-cli
127.0.0.1:6379>
```

FIGURE 2 – Lancement du client redis

1.2 Création et manipulation de variables

On peut désormais commencer à insérer, modifier et supprimer des donnéees depuis le client :

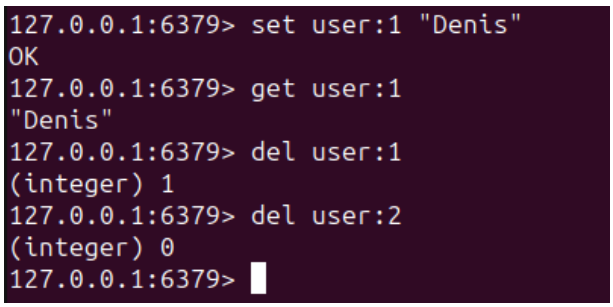

FIGURE 3 – Insertion, affichage et suppression d'utilisateur

On remarque aussi pour chaque commande un retour par OK, pour une insertion réussie mais 1 ou 0 pour del, en fonction du nombre de données supprimées.

On peut aussi modifier des valeurs par exemple en incrémentant une variable :

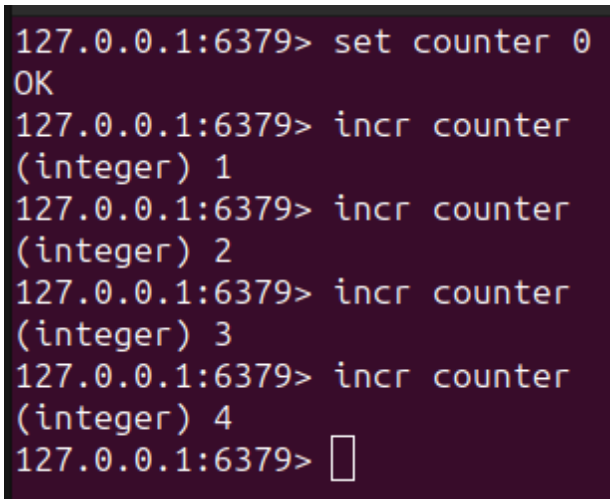

FIGURE 4 – Incrémentation d'un compteur

Il est possible d'avoir des valeurs temporaires :

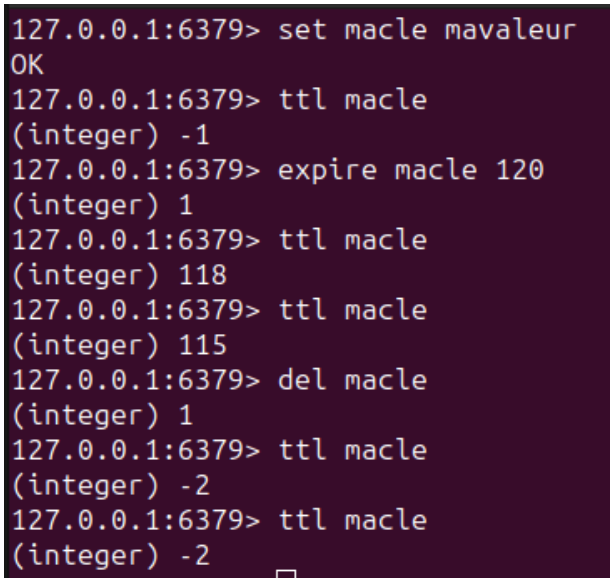

FIGURE 5 – Exemple : macle

La variable macle a par défaut un ttl (time to live) infini exprimé par un -1. on peut le changer et cette donnée expirera après les 120 secondes que l'on a indiqué. (On remarque un ttl de -2 pour toutes les variables n'existant pas ou plus.

1.3 Manipulation de listes

Il est possible de créer des listes et la manière dont elles fonctionne est à la fois comme une pile et une file. On peut y insérer des données par la gauche (LPUSH) ou par la droite (RPUSH) et les sortir par la gauche (LPOP) ou par la droite aussi (RPOP).

Exemple :

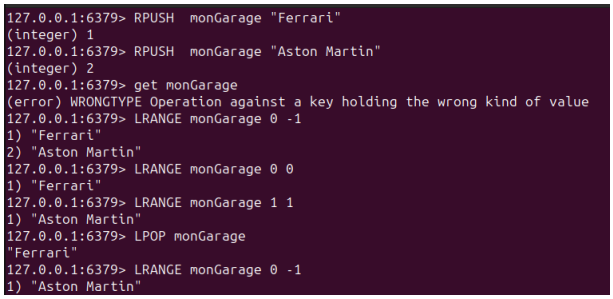

FIGURE 6 – Liste monGarage avec insertion et suppression de données

1.4 Manipulation de Sets

Ensuite il existe les Sets, qui permettent de stocker uniquement des données uniques (SADD set-name value)

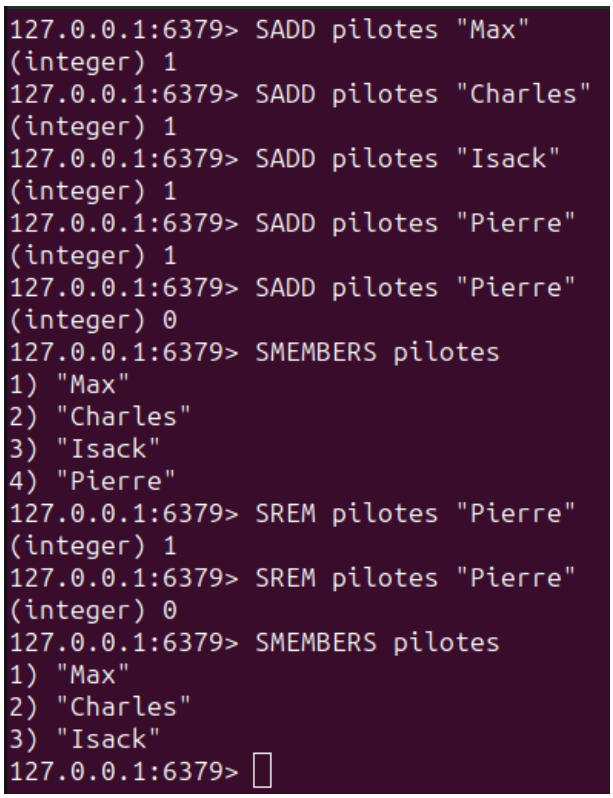

FIGURE 7 – Exemple de Set

Il est possible de faire l'union de deux Sets ce qui permet de récupérer toutes les valeurs uniques de l'union des deux Set.

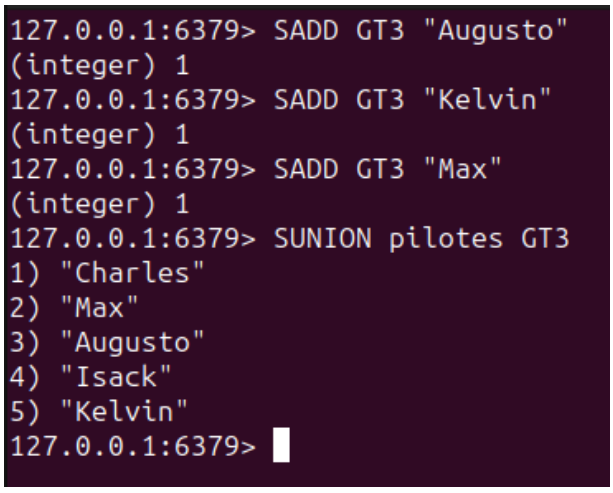

FIGURE 8 – Exemple d'union entre pilotes et GT3 qui contiennent tous les deux Max

2 Vidéo redis 2 : Autres structures de données

2.1 Set ordonnés

Il existe aussi des Set triés que l'on peuple avec ZADD qui permet de trier par ordre croissant les données en fonction d'une valeur.

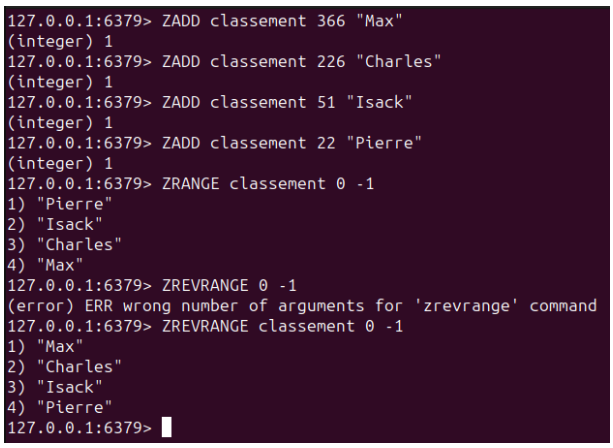

FIGURE 9 – Exemple de Set ordonné

2.2 Manipulation des Hashmap

Il est possible avec redis de stocker avec un clé, plusieurs valeurs associées à d'autres clé. Par exemple, pour un répertoire, on voudrait stocker dans user :1, le nom, l'age et l'email de user :1. Pour ce faire on utilise une hashmap avec la commande HSET.

FIGURE 10 – Exemples de commandes pour les hashmap

3 Vidéo redis 3 : Abonnements et publications

3.1 Abonnements

Il est possible pour 2 clients de communiquer avec des abonnements et publications.

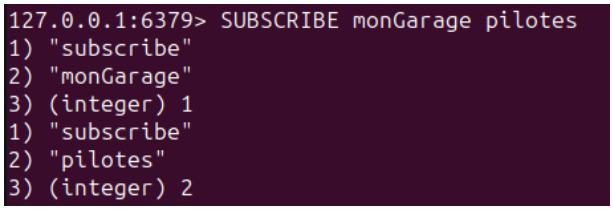

FIGURE 11 – Un deuxième client s'abonne aux cannaux monGarage et pilotes

```
127.0.0.1:6379> PUBLISH monGarage "Nouvelles voitures disponibles!"
(integer) 1
127.0.0.1:6379> PUBLISH pilotes "MAx Verstappen gagne son 5e championnat du monde!"
(integer) 1
127.0.0.1:6379>
```

FIGURE 12 – Lorsque le premier client publie sur l'un des cannaux...

```
1) "message"
2) "monGarage"
3) "Nouvelles voitures disponibles!"
1) "message"
2) "pilotes"
3) "MAx Verstappen gagne son 5e championnat du monde!"
```

FIGURE 13 – ...le deuxième reçoit tout !