Rapport de TP : Introduction pratique à Redis

Nom/Prénom : GNOUG Omayma

Date : 02/12/2025

Sujet : Manipulation des structures de données et administration Redis.

1. Introduction

L'objectif de ce laboratoire est de découvrir et de manipuler les principales structures et commandes de Redis, une base de données NoSQL orientée clé-valeur fonctionnant principalement en mémoire RAM (In-Memory). À travers trois étapes, nous avons exploré les types de données fondamentaux, les structures avancées (Sorted Sets, Hashes) ainsi que le système de messagerie (Pub/Sub).

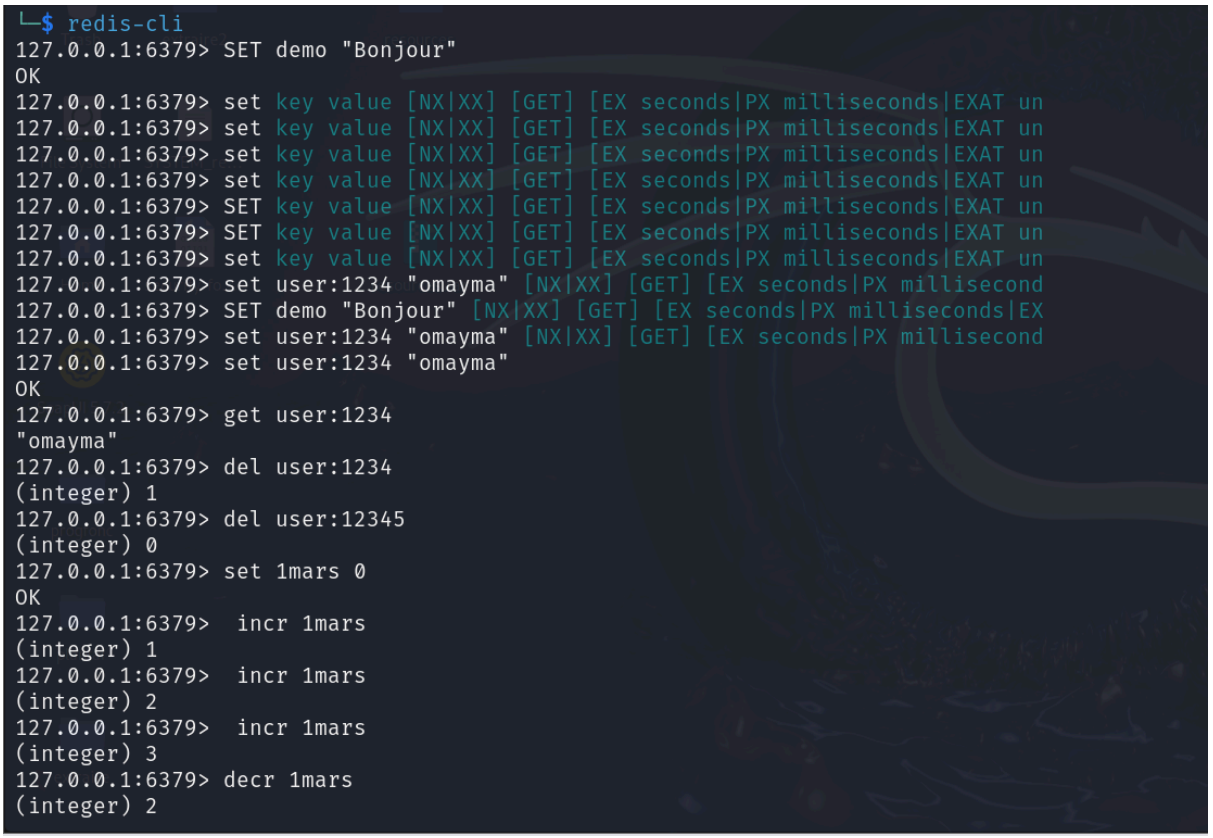
2. Structures de base et commandes essentielles

Dans la première partie, nous avons pris en main les opérations CRUD sur les chaînes de caractères (Strings) et la gestion du cycle de vie des données.

Commandes exécutées sur les Strings :

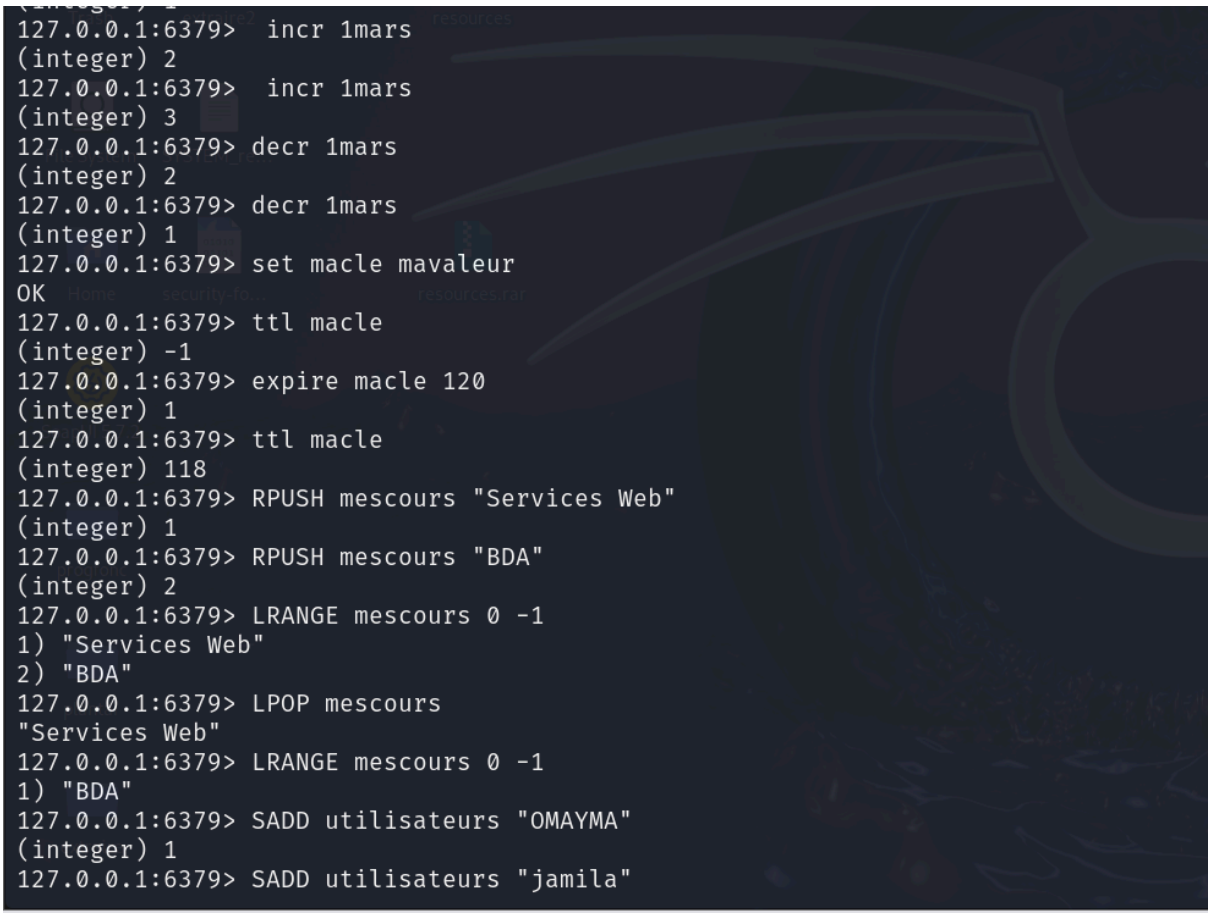
- SET, GET, DEL : Création, lecture et suppression.
- INCR, DECR : Gestion des compteurs atomiques.

- TTL, EXPIRE : Définition d'une durée de vie pour le cache.



Nous avons ensuite manipulé deux structures de collection :

- **Les Listes :** Ajout (LPUSH/RPUSH), retrait (LPOP) et consultation (LRANGE).
- **Les Sets (Ensembles) :** Gestion de valeurs uniques sans ordre défini via SADD, SMEMBERS et SREM.



3. Structures avancées : Sorted Sets et Hashes

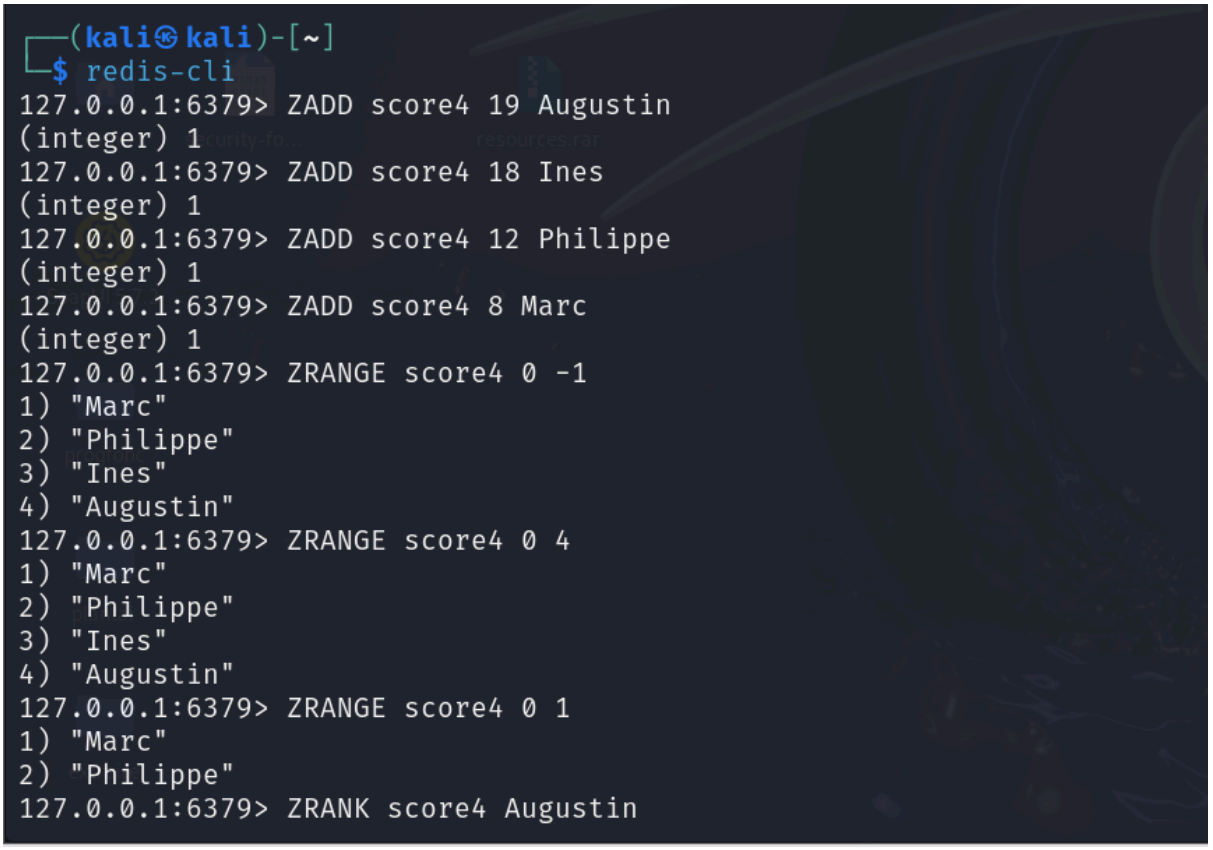
Pour des besoins plus complexes comme les classements ou la structuration d'objets, nous avons utilisé des types de données spécifiques.

Ensembles Ordonnés (ZSET)

Ce type associe un score à chaque membre, permettant un tri automatique.

Commandes :

Code snippet
ZADD mes_scores 100 "Joueur1" // Ajout avec score
ZRANGE mes_scores 0 -1 // Affichage trié
ZRANK mes_scores "Joueur1" // Récupération de la position

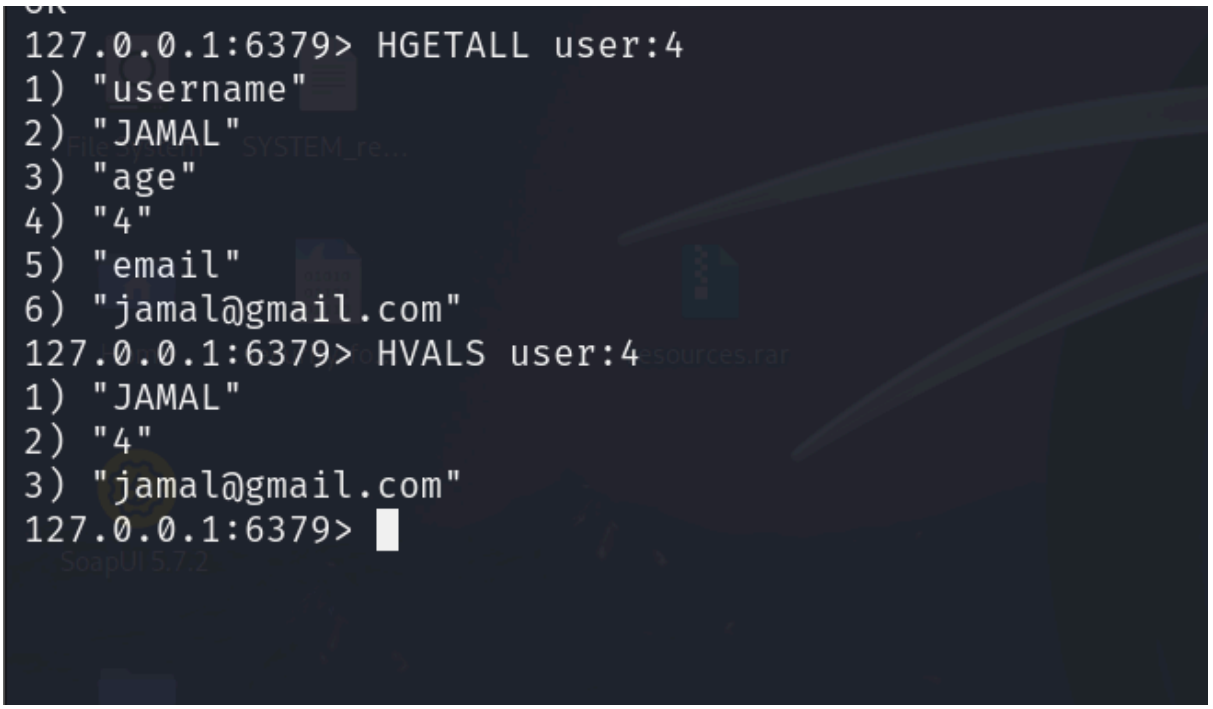


Hashes

Ils permettent de regrouper plusieurs champs sous une même clé, simulant un objet JSON sans schéma strict.

Commandes :

Code snippet
HSET user:1 nom "GNOUG" prenom "Omayma"
HGETALL user:1
HINCRBY user:1 age 1

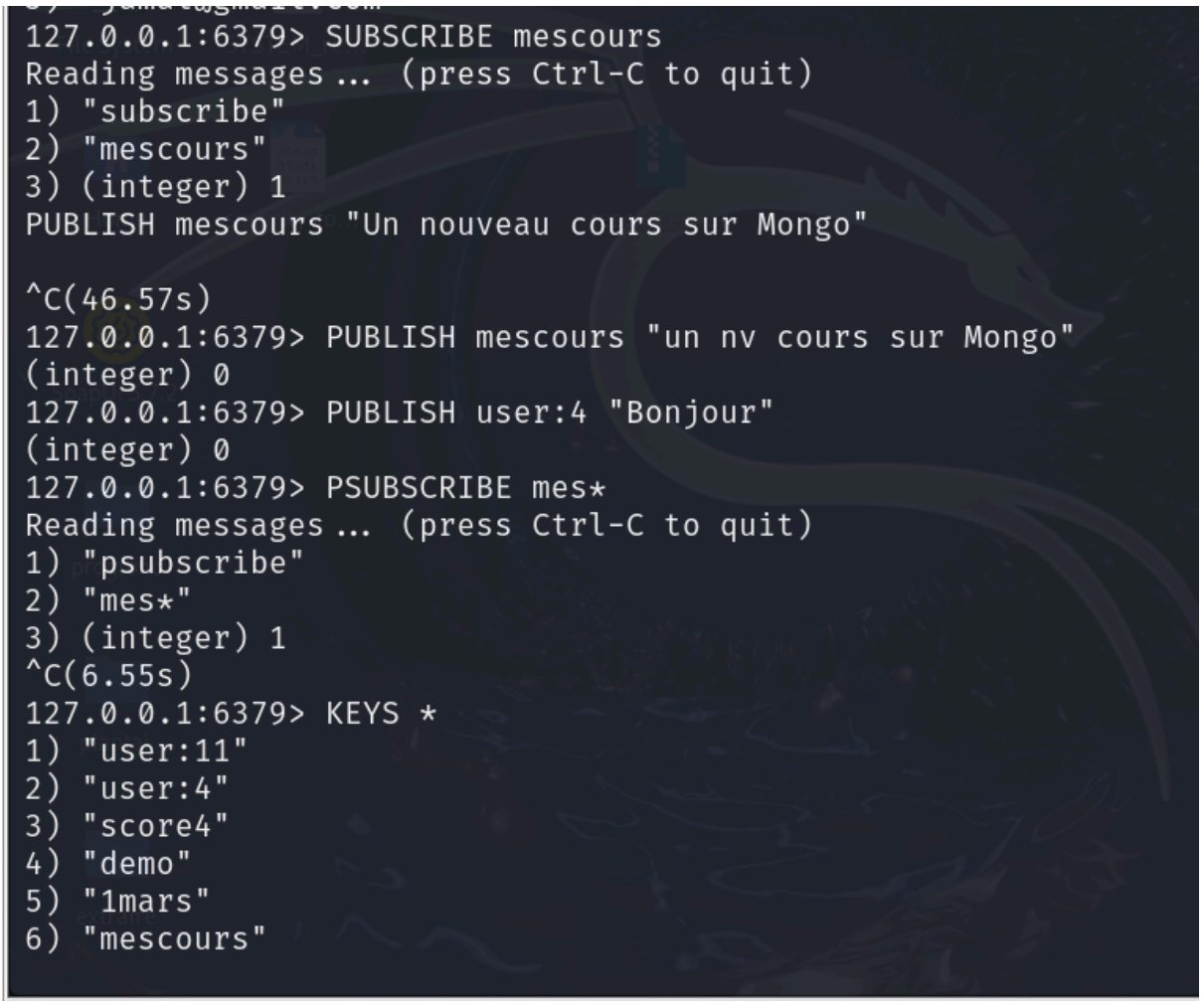


Note : Nous avons observé la différence de performance significative entre l'accès RAM (Redis) et l'accès disque.

4. Système de Publication / Souscription (Pub/Sub)

Nous avons mis en œuvre la fonctionnalité de messagerie temps réel de Redis.

- **Abonnement :** SUBSCRIBE mescours (Le client reste en écoute active).
- **Publication :** PUBLISH mescours "Contenu du message" (Diffusion instantanée aux abonnés).
- **Pattern Matching :** PSUBSCRIBE me* (Abonnement global à tous les canaux commençant par un motif).

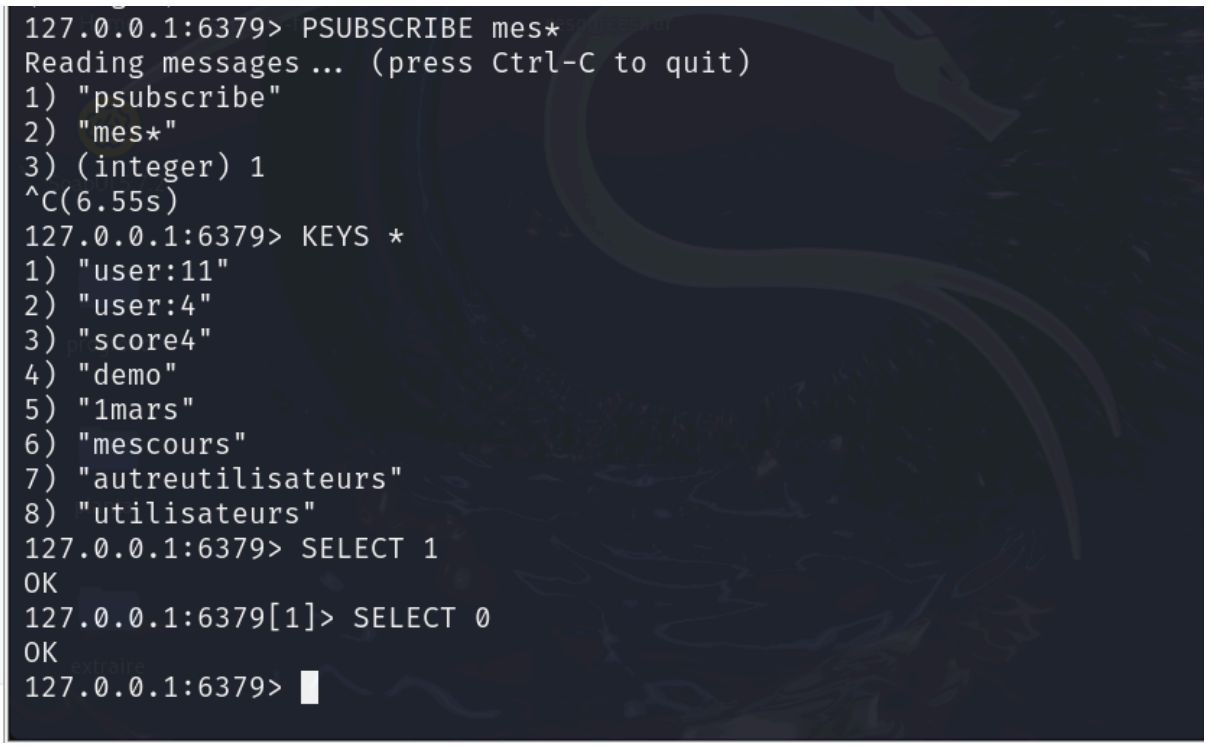


5. Gestion des bases internes

Redis isole les données dans 16 bases logiques (indexées de 0 à 15). Nous avons appris à naviguer entre ces espaces.

Commandes :

Code snippet
SELECT 1 // Basculer vers la base n°1
KEYS * // Afficher les clés de la base active
SELECT 0 // Revenir à la base par défaut



6-Conclusion:

Ce TP nous a permis de comprendre en profondeur les mécanismes fondamentaux de Redis et de manipuler ses principales structures de données. Grâce aux exercices réalisés, nous avons pu expérimenter la gestion de messages en temps réel via Pub/Sub, l'organisation des données au sein de plusieurs bases, et l'utilisation de commandes essentielles pour l'inspection et la modification des clés.
Cette séance met en évidence la puissance et la simplicité de Redis dans les applications modernes, qu'il s'agisse de systèmes distribués, de mise en cache, de gestion d'événements ou de traitement en temps réel. Les compétences acquises au cours de ce TP constituent une base solide pour intégrer Redis dans des projets plus complexes ou orientés performance.