

TP1 – Introduction à Redis

Base de Données NoSQL

Haiballa El Varougou

Table des matières

1	Introduction	3
2	Les quatre familles de bases NoSQL	3
2.1	Bases Clé-Valeur	3
2.2	Bases Documentaires	3
2.3	Bases Orientées Colonnes	3
2.4	Bases Orientées Graphes	3
3	Installation et Déploiement	3
3.1	Installation sous Linux	3
3.2	Installation sous macOS (Homebrew)	3
3.3	Installation via Docker	4
3.4	Déploiement avec Docker Compose	4
4	Premiers Pas avec Redis	4
4.1	Manipulation des clés	4
4.2	Compteurs	4
5	Structures de Données Redis	4
5.1	Listes	4
5.2	Ensembles	5
5.3	Hashmaps	5
5.4	Sorted Sets	5
6	Comprendre les Structures Redis et L/R	5
6.1	Listes et logique Left/Right	5
6.2	Ensembles	5
6.3	Ensembles ordonnés (Sorted Sets)	6
6.4	Hashmaps	6

7	Publication / Souscription (Pub/Sub)	6
7.1	S'abonner	6
7.2	Publier	6
7.3	Abonnement par motif	6
7.4	Autre exemple de publication	6
8	Conclusion	6

1 Introduction

Redis est une base NoSQL orientée clé/valeur extrêmement performante, fonctionnant entièrement en mémoire et supportant diverses structures : listes, ensembles, hashmap, sorted sets, Pub/Sub, etc.

2 Les quatre familles de bases NoSQL

2.1 Bases Clé-Valeur

Stockage simple sous forme `clé → valeur`. Très rapide. Exemples : Redis, Memcached.

2.2 Bases Documentaires

Stockage en documents JSON/BSON. Exemples : MongoDB, CouchDB.

2.3 Bases Orientées Colonnes

Optimisées pour l'analyse Big Data. Exemples : Cassandra, HBase.

2.4 Bases Orientées Graphes

Stockent des noeuds et des relations. Exemples : Neo4j, ArangoDB.

Redis appartient à la catégorie clé/valeur mais propose des structures très avancées.

3 Installation et Déploiement

3.1 Installation sous Linux

```
sudo apt update
sudo apt install redis-server
sudo systemctl enable redis
sudo systemctl start redis
redis-cli
```

3.2 Installation sous macOS (Homebrew)

```
brew update
brew install redis
brew services start redis
redis-cli
```

3.3 Installation via Docker

```
docker pull redis
docker run -d --name redis-tp -p 6379:6379 redis
docker exec -it redis-tp redis-cli
```

3.4 Déploiement avec Docker Compose

Créer un fichier docker-compose.yml :

```
version: "3.8"
services:
  redis:
    image: redis:latest
    container_name: redis-tp
    ports:
      - "6379:6379"
    command: ["redis-server", "--appendonly", "yes"]
    volumes:
      - ./data:/data
```

Puis lancer :

```
docker compose up -d
docker exec -it redis-tp redis-cli
```

4 Premiers Pas avec Redis

4.1 Manipulation des clés

```
SET nom "Alice"
GET nom
SET nom "Bob"
DEL nom
```

4.2 Compteurs

```
SET visiteurs 0
INCR visiteurs
DECR visiteurs
```

5 Structures de Données Redis

5.1 Listes

```
RPUSH cours "Math" "Physique" "Chimie"
```

```
LRANGE cours 0 -1
LPOP cours
```

5.2 Ensembles

```
SADD noms "Alice" "Bob" "Charlie"
SMEMBERS noms
SREM noms "Charlie"
```

5.3 Hashmaps

```
HSET user:1 name "Alice" age 25 email "alice@example.com"
HGETALL user:1
HINCRBY user:1 age 1
```

5.4 Sorted Sets

```
ZADD scores 19 "Augustin"
ZADD scores 18 "Ines"
ZADD scores 20 "Philippe"
ZRANGE scores 0 -1
ZREVRANGE scores 0 -1
ZRANK scores "Augustin"
```

6 Comprendre les Structures Redis et L/R

6.1 Listes et logique Left/Right

Redis utilise :

- L = Left (début)
- R = Right (fin)

```
RPUSH cours "Math" "Physique"
LPOP cours
RPOP cours
```

6.2 Ensembles

Valeurs uniques, sans ordre.

```
SADD noms "Alice" "Bob" "Charlie"
SMEMBERS noms
```

6.3 Ensembles ordonnés (Sorted Sets)

```
ZADD scores 10 "Alice"  
ZADD scores 15 "Bob"  
ZRANGE scores 0 -1
```

6.4 Hashmaps

```
HSET user:1 name "Alice" age 22  
HGET user:1 name
```

7 Publication / Souscription (Pub/Sub)

7.1 S'abonner

```
SUBSCRIBE cours
```

7.2 Publier

```
PUBLISH cours "Un nouveau chapitre est disponible."
```

7.3 Abonnement par motif

```
PSUBSCRIBE cours_*
```

7.4 Autre exemple de publication

```
PUBLISH cours_info "TP Redis disponible."
```

8 Conclusion

Ce TP m'a permis de :

- comprendre les bases de Redis,
- manipuler les structures fondamentales : listes, sets, hashmaps, sorted sets,
- gérer la publication/souscription,
- déployer Redis via Docker et Docker Compose.

Redis est un outil puissant pour le caching, les classements en temps réel, les files de messages et la gestion d'état.