Redis TP 1 : Installation

Prérequis :

- VM centos 7.x
- yum update -y
- Docker

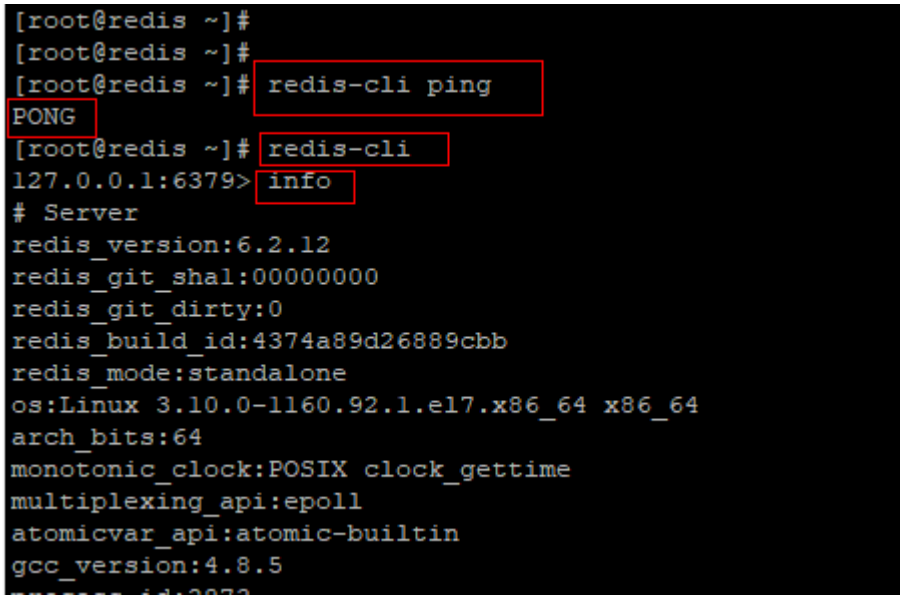
Installation :

```
yum -y install http://rpms.remirepo.net/enterprise/remi-release-7.rpm
yum --enablerepo=remi install redis
rpm -qi redis
```

Lancement du serveur redis :

```
redis-server
```

Validation :



Redis TP 2 : Gestion des données

- Création de données
- Suppression de données
- Test si une key existe
- Création de keys volatiles (avec timeout)
- Suppression de timeout d'une key
- Manipulation de keys dans différentes spaces (bases)

Redis TP 3 : Client Redis

- Install repo epel :

```
yum install epel-release
```

- Install librarie python-redis :

```
yum --enablerepo=epel -y install python-redis
```

- Exécuter le script suivant :

```
#!/usr/bin/env python
import redis

client = redis.StrictRedis(host='127.0.0.1', port=6379, db=0)

# set and get Key
client.set("key01", "value01")
print "key01.value :", client.get("key01")

# append and get Key
client.append("key01", ",value02")
print "key01.value :", client.get("key01")

client.set("key02", 1)

# increment
client.incr("key02", 100)
print "key02.value :", client.get("key02")

# decrement
client.decr("key02", 51)
print "key02.value :", client.get("key02")

# list
client.lpush("list01", "value01", "value02", "value03")
print "list01.value :", client.lrange("list01", "0", "2")

# hash
client.hmset("hash01", {"key01": "value01", "key02": "value02", "key03": "value03"})
print "hash01.value :", client.hmget("hash01", ["key01", "key02", "key03"])

# set
client.sadd("set01", "member01", "member02", "member03")
print "set01.value :", client.smembers("set01")
```

Redis TP 4 : Réplication

- Arrêter le service redis.service
- Lancer deux instances Redis server sur deux consoles Shell

Sur console 1 exécuter :

```
redis-server --port 6379 --dbfilename db1.rdb
```

Sur console 2 exécuter :

```
redis-server --port 6380 --dbfilename db2.rdb
```

Connexion aux deux instances :

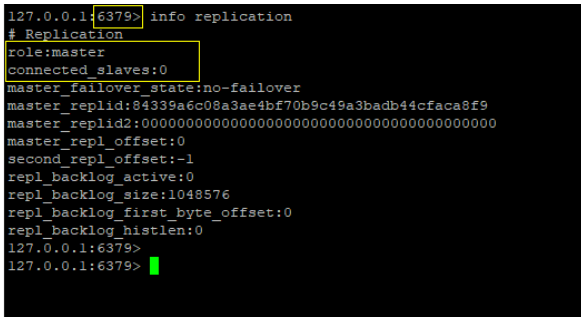
Sur console 3 exécuter **:**

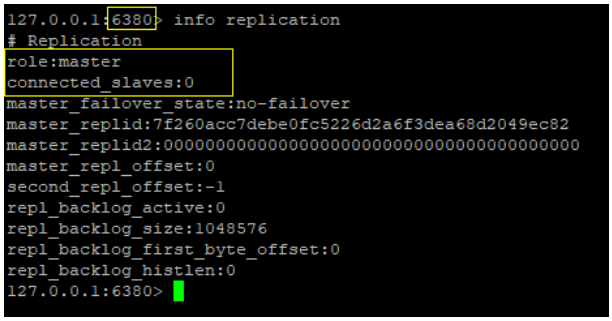
```
redis-cli -p 6379
puis keys * (pour vérifier l'absence de données)
```

Sur console 4 exécuter :

```
redis-cli -p 6380
puis keys * (pour vérifier l'absence de données)
```

Sur les consoles 3 et 4 exécuter :

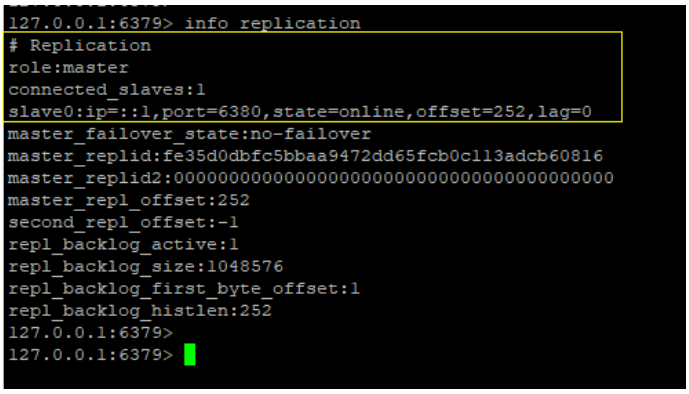


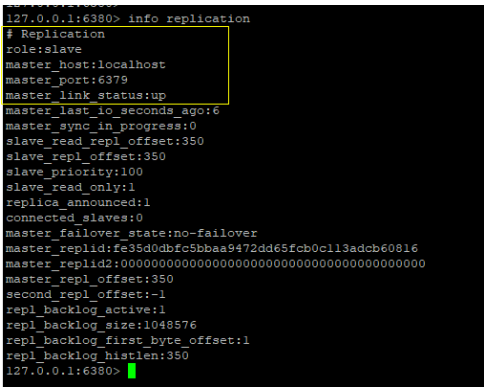


Sur la console 4 (port 6380), exécuter la commande :

```
REPLICAOF localhost 6379
```

Sur les consoles 3 et 4 exécuter



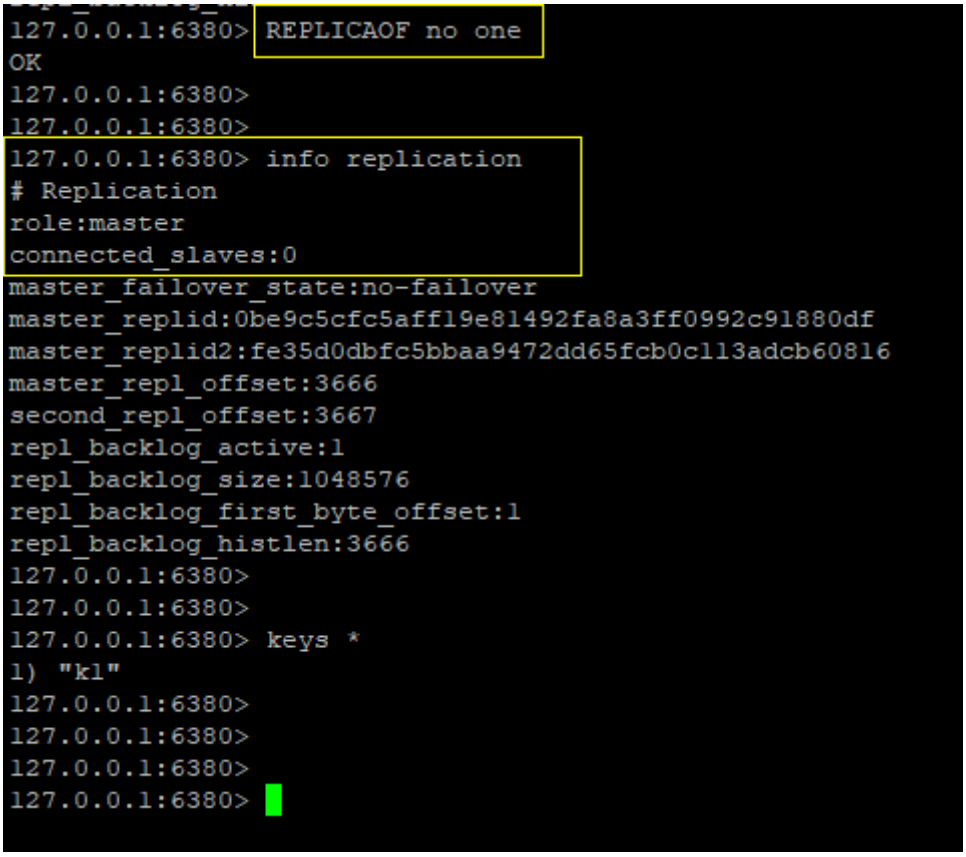


Sur le master : créer des données

Sur le slave : afficher les données créées sur le master

Sur le slave : essayer de créer des données. Noter le retour de la commande de création.

Sur le slave exécuter : replicaof no one



Redis TP 5 : Cluster

- Création de 6 répertoires : 7001, 7002, 7003, 7004, 7005, 7006
- Création des fichiers :

n1_redis.conf dans 7001
n2_redis.conf dans 7002
n3_redis.conf dans 7003
n4_redis.conf dans 7004
n5_redis.conf dans 7005
n6_redis.conf dans 7006

```
port 700X
cluster-enabled yes
cluster-config-file nodes.conf
cluster-node-timeout 5000
appendonly yes
```

X : 1, 2, 3, 4, 5, 6

- Dans chaque répertoire exécuter la commande :

```
redis-server nX_redis.conf &
```

- Ala fin des opérations, vérifier que tous les serveurs sont bien lancés :

```
[root@redis 7006]# ps aux | grep redis | grep -v grep
root      2577  0.1  1.0 165596 10268 pts/0    Sl   16:13   0:00 redis-server *:7001 [cluster]
root      2594  0.1  1.0 162524 10276 pts/0    Sl   16:13   0:00 redis-server *:7002 [cluster]
root      2610  0.1  1.0 162524 10272 pts/0    Sl   16:14   0:00 redis-server *:7003 [cluster]
root      2626  0.0  1.0 162524 10272 pts/0    Sl   16:14   0:00 redis-server *:7004 [cluster]
root      2642  0.0  1.0 162524 10272 pts/0    Sl   16:14   0:00 redis-server *:7005 [cluster]
root      2659  0.0  1.0 162524 10276 pts/0    Sl   16:15   0:00 redis-server *:7006 [cluster]
[root@redis 7006]#
```

- Lancer la commande de création du cluster :

```
redis-cli --cluster create 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004
127.0.0.1:7005 127.0.0.1:7006 --cluster-replicas 1
```

```
[root@redis ~]# redis-cli --cluster create 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004 127.0.0.1:7005 127.0.0.1:7006 --cluster-replicas 1
>>> Performing hash slots allocation on 6 nodes...
Master[0] -> Slots 0 - 5460
Master[1] -> Slots 5461 - 10922
Master[2] -> Slots 10923 - 16383
Adding replica 127.0.0.1:7005 to 127.0.0.1:7001
Adding replica 127.0.0.1:7006 to 127.0.0.1:7002
Adding replica 127.0.0.1:7004 to 127.0.0.1:7003
>>> Trying to optimize slaves allocation for anti-affinity
[WARNING] Some slaves are in the same host as their master
M: dfd393b9ceel302f27ac0dd443d4bc3f19859d35 127.0.0.1:7001
   slots:[0-5460] (5461 slots) master
M: a64ed5fac0691357a8a5b7f1987cf97bac20c040 127.0.0.1:7002
   slots:[5461-10922] (5462 slots) master
M: 6a09270e1494a0445d9cfb6460aa33258624e795 127.0.0.1:7003
   slots:[10923-16383] (5461 slots) master
S: 6db399ada15c23b25d695de35d1d41d40e55e19b 127.0.0.1:7004
   replicates a64ed5fac0691357a8a5b7f1987cf97bac20c040
S: bfd1f9baa7d6fa8bcdbd0e2f49e670c6c557bd2d 127.0.0.1:7005
   replicates 6a09270e1494a0445d9cfb6460aa33258624e795
S: 9ff6aa0f7c512470f4a28f823ba1fd1b9d07a479 127.0.0.1:7006
   replicates dfd393b9ceel302f27ac0dd443d4bc3f19859d35
Can I set the above configuration? (type 'yes' to accept):
```

- Répondre : yes

```
2659:S 26 Oct 2023 17:26:51.970 * Residual parent diff successfully flushed to the rewritten AOF (0.00 MB)
2659:S 26 Oct 2023 17:26:51.970 * Background AOF rewrite finished successfully
2626:S 26 Oct 2023 17:26:51.972 * Background AOF rewrite terminated with success
2626:S 26 Oct 2023 17:26:51.972 * Residual parent diff successfully flushed to the rewritten AOF (0.00 MB)
2626:S 26 Oct 2023 17:26:51.972 * Background AOF rewrite finished successfully
2642:S 26 Oct 2023 17:26:51.978 * Background AOF rewrite terminated with success
2642:S 26 Oct 2023 17:26:51.978 * Residual parent diff successfully flushed to the rewritten AOF (0.00 MB)
2642:S 26 Oct 2023 17:26:51.978 * Background AOF rewrite finished successfully
2594:M 26 Oct 2023 17:26:54.791 # Cluster state changed: ok
2577:M 26 Oct 2023 17:26:54.792 # Cluster state changed: ok
2610:M 26 Oct 2023 17:26:54.792 # Cluster state changed: ok
```

- Se connecter au cluster (avec l'argument -c en plus)

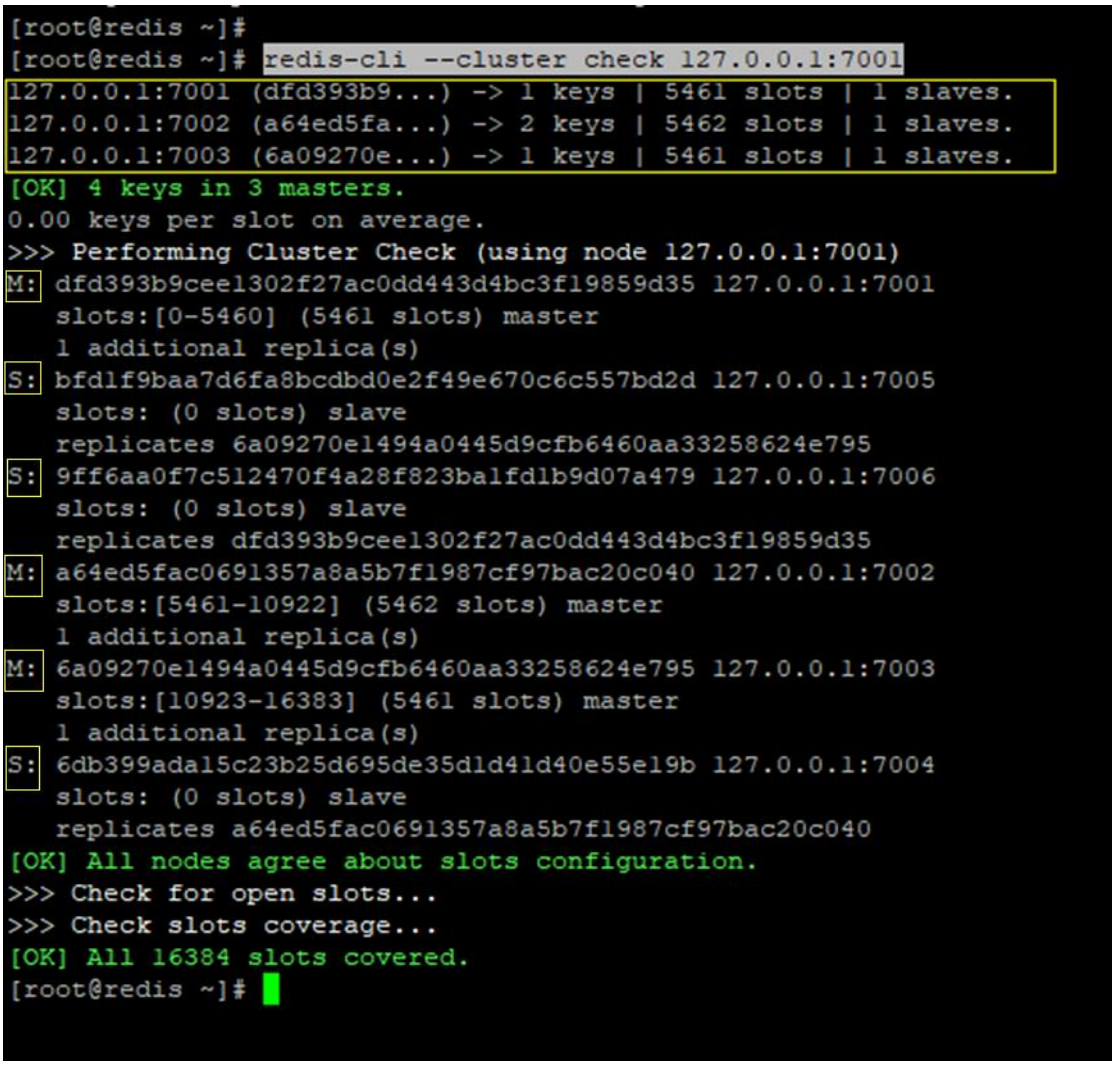
Redis-cli -c -p 7001

```
127.0.0.1:7001> KEYS *
(empty array)
127.0.0.1:7001>
127.0.0.1:7001> set name doe
-> Redirected to slot [5798] located at 127.0.0.1:7002
OK
127.0.0.1:7002> get name
"doe"
127.0.0.1:7002>
127.0.0.1:7002> set kl vall
-> Redirected to slot [12706] located at 127.0.0.1:7003
OK
127.0.0.1:7003>
127.0.0.1:7003> get kl
"vall"
127.0.0.1:7003>
127.0.0.1:7003> get name
-> Redirected to slot [5798] located at 127.0.0.1:7002
"doe"
127.0.0.1:7002>
127.0.0.1:7002>
```

- Se connecter à un nœud Slave et vérifier s'il a répliqué les données de son Master
- Sur le Slave, créer une key (set … …)
- Que peut-on constater ?

- Depuis le Shell exécuter la commande :

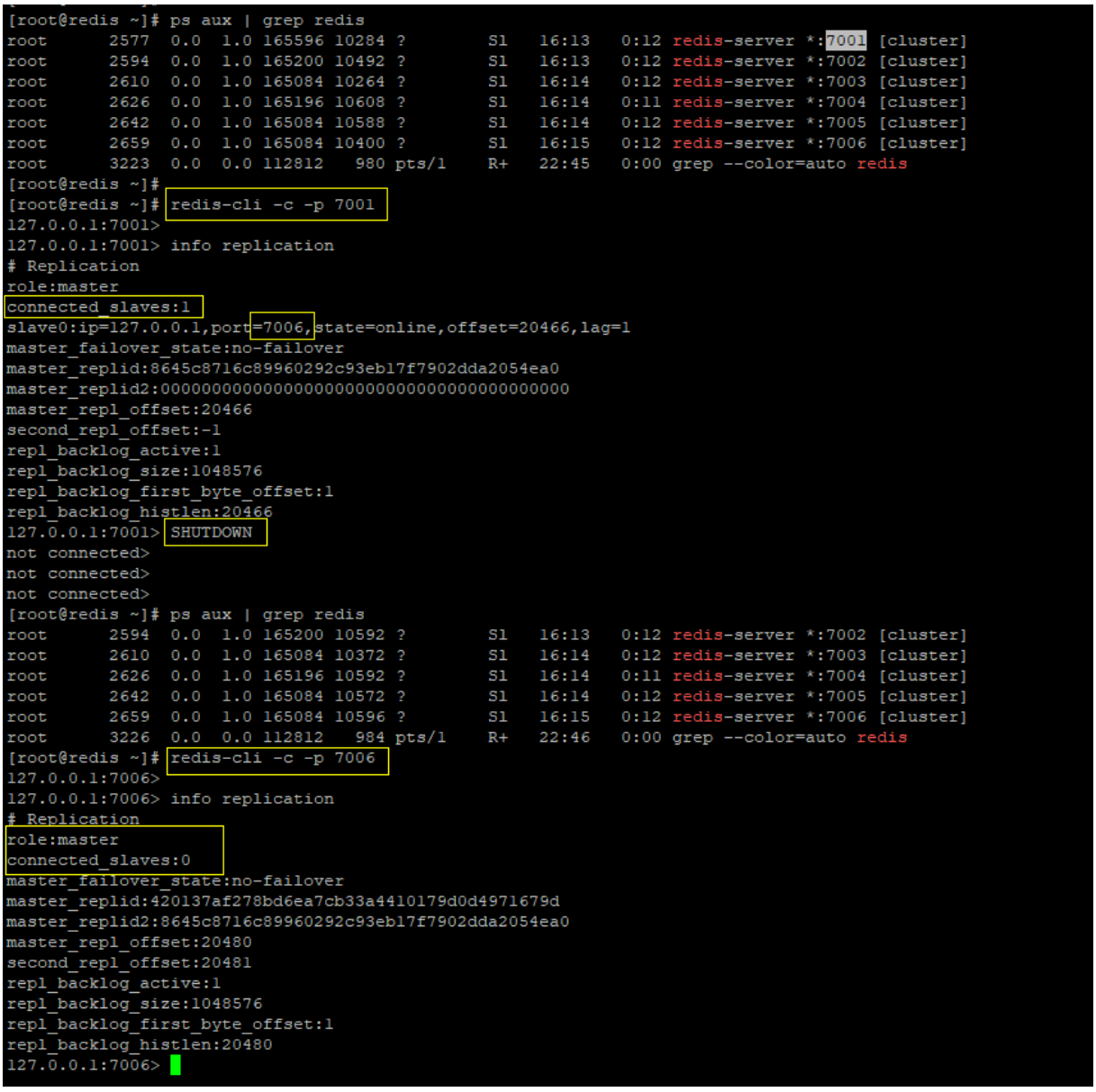
```
redis-cli --cluster check 127.0.0.1:7001
```



- Vérifier le nombre de keys dans chaque Master.

Redis TP 6 : Haute disponibilité

- Se connecter à un master : 7001 par exemple
- Noter le port de son slave --> 7006
- Arrêter le master --> shutdown
- Se connecter au Slave
- Exécuter la commande : info



- Relancer le serveur qui a été arrêté --> 7001, puis s'y connecter
- Exécuter la commande : info, ou la commande role
- Que peut-on remarquer ?

Redis TP 7 : RediSearch

- Lancer un container RediSearch par la commande :

```
docker run -p 6379:6379 redislabs/redisearch:latest
```

- Sur une autre console se connecter à redisearch :

```
Redis-cli -p 6379
```

Redis TP 8 : RediSearch – Indexation & Requêtes

- Insertion des données dans redis :

Download les fichiers de données depuis gitlab.com : https://gitlab.com/ctouil/redis.git

```
Redis-cli < import_movies.redis
```

- Création d'index :

```
ft.create idx:movie ON hash PREFIX 1 "movie" SCHEMA movie_name TEXT SORTABLE release_year NUMERIC SORTABLE rating NUMERIC SORTABLE category TAG SORTABLE
```

Requête de recherche :

- **Recherche de tous les films contenant le mot « war »**

```
FT.SEARCH idx:movie "war"
```

Combien de résultats ?

- **Recherche de tous les films contenant le mot « war » mais pas le mot jedi**

```
FT.SEARCH idx:movie "war -jedi"
```

- **Recherche avec affichage de certains champs** : même requête en affichant uniquement movie_name et release_year

```
FT.SEARCH idx:movie "war" RETURN 2 movie_name release_year
```

RETURN Nombre_de_champs champ1 champ2

- **Recherche dans un champ bien défini** : recherche du mot war dans le champ movie_name

```
FT.SEARCH idx:movie "@movie_name:war" RETURN 2 movie_name release_year
```

- **Recherche basée sur les valeurs** : recherche des films dont "category" = thriller

```
FT.SEARCH idx:movie "@category:{thriller}" RETURN 3 movie_name release_year category
```

- **Recherche de plusieurs valeurs** : recherche des films dont "category" = « thriller » ou « action »

```
FT.SEARCH idx:movie "@category:{thriller|action}" RETURN 3 movie_name release_year category
```

- **Recherche sur plusieurs champs** : recherche des films dont "category" = « thriller » ou « action » et movie_name ne contient pas le mot « jedi »

```
FT.SEARCH idx:movie "@category:{thriller|action} @movie_name:-jedi" RETURN 3 movie_name release_year category
```

- **Recherche dans une plage de valeurs** : recherche de films dont release_year est entre 1970 et 1980

```
FT.SEARCH idx:movie "@release_year:[1970 1980]" RETURN 2 movie_name release_year
```

- **Recherche avec affichage ordonné des résultats :**

```
ft.search idx:movie "@category:{Action}" SORTBY release_year return 2 movie_name release_year
```

```
ft.search idx:movie "@category:{Action}" SORTBY release_year DESC return 2 movie_name release_year
```

- **Recherche avec limitation d'affichage :** Par défaut seuls les **10** premiers résultats sont affichés.

```
ft.search idx:movie "@category:{Action}" SORTBY release_year DESC return 2 movie_name release_year LIMIT 0 5
```

```
ft.search idx:movie "@category:{Action}" limit 0 0
```

LIMIT 0 0 : donne le nombre d'enregistrement

Redis TP 9 : RedisInsight

- **Lancer le container RediSearch.**
- **Installer RedisInsight**
- **Explorer les possibilités de l'outil**

Redis TP 10 : Sécurité

- **Activer l'authentification dans une instance Redis**