TP Redis - Replication / Sharding

EISTI

3 février 2019

1 Sharding et replication avec Redis Cluster

Nous allons utiliser Redis cluster pour réaliser une gestion distribuée de partitionnement et réplication.

- Dans le fichier de configuration `redis.conf`, positionner les variables `appendonly` et `cluster-enabled` à yes et `cluster-node-timeout` à 5000.
- Lancer le logiciel en déclarant les master et leurs esclaves (accepter la conf par défaut) :
  `redis-cli --cluster create`
  `$host1:$port1 $host2:$port2 ... --cluster-replicas n`
  La variable n correspond aux nombre de réplicats par noeud maître.
- Quelques commandes de gestion :
  - `redis-cli --cluster check $host:$port` : vérifie le statut ;
  - et d'autres (add-node, del-node, ...)
- Nous allons tester plusieurs insertions sur un noeud maître choisi au hasard (redis-cli -c) :
  - `set util-01 toto` ;
  - `mset util-02 tata util-03 titi` ;
  - `hset set-01 util:id 01 util:pass eisti util:niv CPI1` ;
  - `hset set-02 util:id 02 util:pass eisti util:niv ING1` ;
  - `lpush lusers 01 02` ;
  - `get util-01` ;
  - ...

2 Replication avec Sentinel

L'objectif est de créer un cluster de réplication contenant un noeud master et 3 noeuds slaves.
Le fichier `sentinel.conf` contient les paramètres de l'observateur :

- port : port d'écoute de sentinel
- bind : 0.0.0.0 : pour être accessible
- protected-mode no

— `sentinel monitor <master-name> <ip> <redis-port> <quorum>`.
  — `master-name` : nom commun au noeud master et aux noeud slaves ;
  — `quorum` : nombre d'instances redis nécessaires pour constater qu'un noeud master est tombé ;
— `sentinel down-after-milliseconds <master-name> <milliseconds>` : temps après lequel le noeud est considéré comme tombé et la procédure de failover est lancée (nous prendrons 5000).
— Quelques commandes de gestion :
  `redis-cli -p $port -h $host sentinel <options/valeurs>`
  — masters : récupère les noeuds maîtres et leur statut ;
  — master <master-name> : récupère le noeud maître ;
  — slaves : récupère les noeuds esclaves et leur statut ;
  — sentinels : récupère les instances sentinelles ;
  — et d'autres (chkquorum, failover, ...)

Nous allons créer 4 sentinelles surveillant 4 serveurs.

— Lancement des sentinelles : `redis-sentinel path-to/sentinel.conf`
  ou
  `redis-server path-to/sentinel.conf --sentinel`
— Pour connaître le noeud master, on se connecte à une sentinelle et on lance la reqête :
  `redis-cli -h <host> -p <port> sentinel get-master-addr-by-name <master-name>`.