

Redis -NoSQL database



Contents

- 1 Bemutató
- 2 Telepítés
- 3 Konfiguráció
 - ◆ 3.1 Hálózati beállítások
 - ◆ 3.2 Memória használat
 - ◆ 3.3 Adatbázis mentése
 - ◆ 3.4 Adatbázisok száma
 - ◆ 3.5 Indítás
- 4 Redis monitorozása és failover
- 5 Webes admin felület (phpRedisAdmin)
 - ◆ 5.1 SELinux
- 6 Backup és restore
- 7 Kliensek
 - ◆ 7.1 PHP kliens
 - ◆ 7.2 Java kliens

Bemutató

Redis 3.2.5 - a legújabb stabil verzió a cikk írásakor

A Redis egy memória alapú adatstruktúra adatbázis. Nagyon sokféle struktúrát támogat, nem csak string-eket. (Set, Map, Hash, List). Az elosztott memóriát is támogatja, és képes lemeze menteni a memória cache tartalmát.

Ezt írják magukról:

"Redis is an open source (BSD licensed), in-memory data structure store, used as database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs and geospatial indexes with radius queries. Redis has built-in replication, Lua scripting, LRU eviction, transactions and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster."

Telepítés

A Redis az EPEL repozitóriumban van benne, ezt hozzá kell adni a YUM-hoz ha még nem lenne benne:

```
# wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
# rpm -ivh epel-release-6-8.noarch.rpm
```

Majd telepítsük föl a Redis-t.

```
# yum install redis
```

Létrejött fájlok:

- /etc/redis.conf: A Redis központi konfigurációs fájlja, mi is ebben fogjuk a szükséges konfigurációt elvégezni.
- /etc/redis-sentinel.conf: A Redis központi monitorozó eszköze. Figyeli az esetlegesen fellépő hibákat, és megpróbál automatikusan beavatkozni.
- /var/log/redis: Log fájl helye
- /etc/logrotate.d/redis: A logrotate-thez is hozzáadta a telepítést a szükséges konfigurációt.
- /var/lib/redis: Ide fogja kiírni a Redis a memóriában összegyűjtött adatbázist a konfigurációban meghatározott gyakorisággal.

Konfiguráció

Hálózati beállítások

Amennyiben lehetséges, a **NETWORK** állítsuk be, hogy a Redis csak a lokális hálózaton hallgatózzon és csak egy porton biztonsági megfontolásból. /etc/redis.conf

```
bind 127.0.0.1
port 6379
```

Memória használat

Mivel cache-ként szeretnénk használni a Redis-t, és nem adatbázisként, maximitálni fogjuk, hogy mennyi memóriát használhat el a rendszermemóriából, mivel nem történik tragédia, ha egy új cache bejegyzés már nem fér bele a memóriába.

<http://redis.io/topics/lru-cache>

A **LIMITS** szekcióban állítjuk be, hogy mennyi memóriát használhat maximálisan a redis, és hogy mit tegyen, milyen policy-t alkalmazzon, ha már nincs szabad hely. 6 féle memória kezelési stratégia közül választhatunk. Mi a **allkeys-lru** stratégiát fogjuk használni, ami egy algoritmus alapján megkeresi a legkevésbé használt kulcsokat, és azokat kitörli. A többi stratégiával ellentétben ez mindig kitöröl egy kulcsot, ha a memória már megtelt és új INSERT kérés érkezik.
/etc/redis.conf

```
maxmemory 5GB
maxmemory-policy allkeys-lru
```

Adatbázis mentése

A memóriában lévő adatbázis mentésével kapcsolatos beállítások a SNAPSHOTTING szekcióban vannak a redis.conf fájlban. A save paraméterrel adhatjuk meg hogy hány másodperc és hány módosítás elteltével írja ki lemezre a memória cache tartalmát a redis:

```
save 900 1
save 300 10
save 60 10000
```

Mind két paraméternek teljesülnie kell, tehát az első esetben 900 másodperc és legalább 1 módosítás után lementi a cache-t lemezre.

Az adatbázis fájl neve és helye:

```
dbfilename dump.rdb
dir /var/lib/redis/
```

Próbáljuk ki. Adjuk ki a redis-cli parancssori eszköznek a save parancsot, ekkor ki kell hogy írja a memória tartalmát a lemezre:

```
# redis-cli save
OK
```

Láthatjuk, hogy a /var/lib/redis/ mappában a dump.rdb dátuma frissült:

```
# ll /var/lib/redis
-rw-r--r--. 1 redis redis 4894 Nov  6 18:24 dump.rdb
```

Adatbázisok száma

Az alap konfiguráció 16 adatbázist hoz létre, a kliensek 0-tól 15-ig hivatkozhatnak számmal ezekre az adatbázisokra. A 0. adatbázis az alapértelmezett.

```
databases 16
```

Majd a webes admin GUI-ban látni fogjuk, hogy 16 adatbázis közül választhatunk.

Indítás

```
# service redis start
# chkconfig redis on
```

Redis monitorozása és failover

A redis szerverek monitorozására a **redis-sentinel** program szolgál, ami feltelepült a redis yum installációjával együtt.

<http://redis.io/topics/sentinel>

Valójában nagy, klászteres környezetben van értelme a sentinel használatának.

A /etc/redis-sentinel.conf fájlban található alap beállítások megfelelnek. A localhost-on a 6379-es porton fog csatlakozni a redesi szerverhez.

```
sentinel monitor mymaster 127.0.0.1 6379 2
```

Indítsuk el:

```
# service redis-sentinel start
Starting redis-sentinel: [ OK ]

# chkconfig redis-sentinel on
```

Webes admin felület (phpRedisAdmin)

Nagyon sok féle Redis grafikus admin eszköz létezik, ezek között vannak desktop alkalmazások és különféle webes eszközök. Mi egy nagyon egyszerűt, és elterjedt webes GUI-t fogunk használni a **phpRedisAdmin**-t.

A phpRedisAdmin-t git-ből lehet letölteni két lépésben. Lépjünk bele abba apache által látott mappába, ahova a phpRedisAdmin-t telepíteni szeretnénk, majd az alábbi két git repót kell klonozzuk:

```
# cd /var/www/virtual/admin
# git clone https://github.com/ErikDubbelboer/phpRedisAdmin.git
# cd phpRedisAdmin
# git clone https://github.com/nrk/predis.git vendor
```

Nevezzük át a sample konfigurációs fájlt:

```
# cp includes/config.sample.inc.php includes/config.inc.php
```

phpRedisAdmin

local server ▾

database 1 ▾



+ Add another key

*

Filter!

type here to filter

Info

Key	Value
Server	<pre>Array ([redis_version] => 3.2.5 [redis_git_sha1] => 00000000 [redis_git_dirty] => 0 [redis_build_id] => 24e1b5cc5266a2bd [redis_mode] => standalone [os] => Linux 2.6.32-504.el6.x86_64 x86_64 [arch_bits] => 64 [multiplexing_api] => epoll [gcc_version] => 4.4.7 [process_id] => 8426 [run_id] => 7f35008dfcde961913f190b73beef9d698bc479c [tcp_port] => 6379 [uptime_in_seconds] => 68529 [uptime_in_days] => 0 [hz] => 10 [lru_clock] => 2058796 [executable] => /usr/bin/redis-server [config_file] => /etc/redis.conf)</pre>
Clients	<pre>Array ([connected_clients] => 1 [client_longest_output_list] => 0 [client_biggest_input_buf] => 0 [blocked_clients] => 0)</pre>
	<pre>Array ([used_memory] => 823584 [used_memory_human] => 804.28K [used_memory_rss] => 1851392)</pre>

SELinux

A httpd az alap rendszabás szerint nem olvashatja a 6379 portot. Ezért megváltoztatjuk a port típusát. Erre amúgy is szükség lenne, ha php klienssel akarnánk a Redis-hez kapcsolódni.

```
# semanage port -a -t http_port_t -p tcp 6379
```

Backup és resotre

Kliensek

PHP kliens

A php klienst YUM-al tudjuk feltelepíteni.

```
# yum install php-pecl-redis
```

Ekkor a /etc/php.d/ mappába létrejön a 50-redis.ini fájl, ami include-álja a redis.so:

```
extension = redis.so
```

Java kliens