Key

ردیس یک دیتابیس nosql است که اطلاعات را به شکل کلید/مقدار ذخیره میکند. ردیس از انواع مختلفی از دیتاتایپ ها پشتیبانی میکند. ساده ترین نوع داده در ردیس string است. برای ایجاد کردن یک کلید میتوانید از دستور SET و برای خواندن اطلاعات یک کلید از دستور GET استفاده کنید.کلیدها در ردیس binary safe هستند یعنی میتوانند هرگونه دنباله باینری را ذخیره کنند. از یک string ساده گرفته تا محتویات یک عکس. همچنین stringهای خالی هم یک کلید معتبر هستند.

با دستور redis-cli وارد محیط redis می‌شویم

127.0.0.1:6379

===========================================================

string

stringها در ردیس ساده ترین نوع داده در ردیس هستند. stringها در ردیس binary safe هستند یعنی میتوانند هرگونه دنباله باینری را ذخیره کنند. از یک آبجکت زبان پایتون گرفته تا محتویات یک عکس. حداکثر حجم هر string میتواند 512 مگابایت باشد. تمامی اعداد در ردیس به عنوان string ذخیره میشوند. شما میتوانید از دستورات ردیس برای تغییر دادن stringها استفاده کنید.

===========================================================

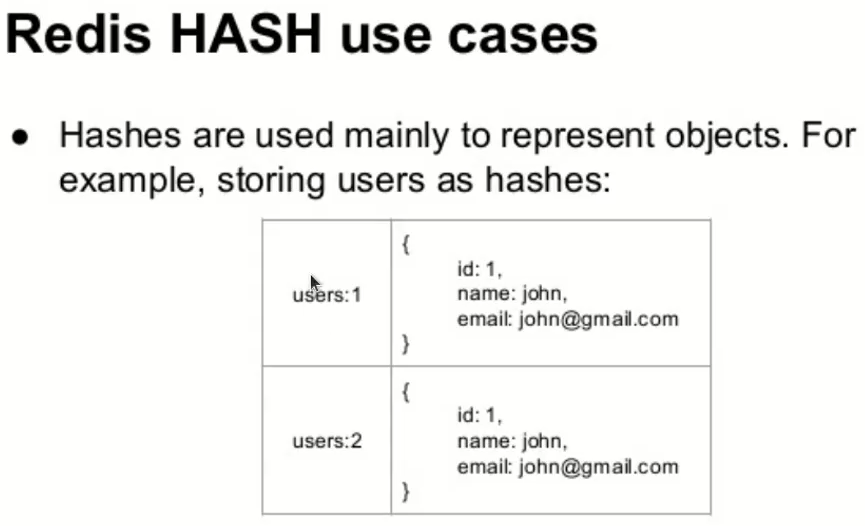
list

با استفاده از listها میتوانید چندین مقدار را داخل یک کلید ذخیره کنید. المان های لیست به ترتیب ورود طبقه بندی میشوند یعنی ترتیب ورود اطلاعات مهم است. هر لیست میتواند بیشتر از ۴ میلیارد المان در خود ذخیره کند. همچنین میتوانید با متدهایی که ردیس در اختیار شما قرار میدهد لیست های خود را تغییر بدهید.

===========================================================

hash

هش ها نوعی از داده هستند که میتوانید در یک کلید چندین مقدار به شکل کلید/مقدار ذخیره کنید. در ردیس هش ها توانایی ذخیره بیش از ۴ میلیادر المان را دارند. هش هایی با بیش از هزار فیلد به شکلی ذخیره میشوند که فضای کمی را اشغال کنند پس شما میتوانید جندین میلیون رکورد را بدون مشکل ذخیره کنید. بیشترین استفاده از هش ها در redis برای دخیره آبجکت ها است



دستورات:

HSET , HGET , HSETNX , HMSET , HDEL , DEL , HEXIST ,

HGETALL , HINCRBY , HINCRBYFLOAT , HKEYS , HVALS , HLEN ,

HMGET , HSTRLEN ,

===========================================================

set در ردیس مجموعه ای از المان هایی است که به شکل نامنظم در یک کلید ذخیره میشوند. با استفاده از متدهایی که redis در اختیار ما قرار داده امکان اضافه کردن و حذف کردن المان ها وجود دارد. setها در ردیس اجازه ذخیره المان های تکراری را نمیدهد و اضافه کردن چندباره المان ها تغییری در تعداد نخواهد داشت. همچنین ردیس از عملیات ها ریاضی مثل اجتماع، اشتراک و تفاوت ها پشتیبانی میکند. هر set امکان ذخیره بیش از ۴ میلیارد المان را دارد

دستورات :

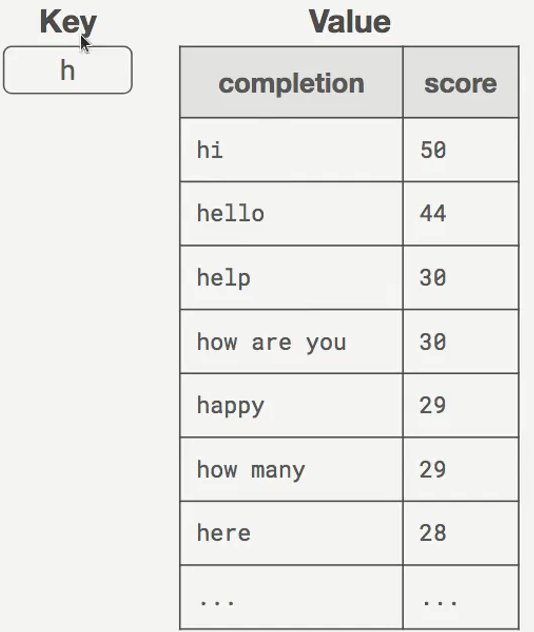
SADD , SMEMBERS , SCARD , SISMEMBER , SPOP , SRANDMEMBER ,

SREM , SMOVE , SDIFF , SINTER , SUNION , SDIFFSTORE

===========================================================

نوع داده sorted set دقیقا مشابه setها است و نمیتوان المان های تکراری در آن ذخیره کرد. تنها تفاوتی که این دو نوع با هم دارند اینست که المان ها در

sorted set یک score یا امتیاز دارند که ردیس المان ها را به ترتیب این scoreها ذخیره میکند. در این نوع داده المان ها نمیتوانند موارد تکراری داشته باشند اما امتیازها میتوانند تکرار شوند. در sorted setها اعمال خواندن، اضافه کردن، حذف کردن و آپدیت کردن المان ها بسیار سریع است



دستورات:

ZADD:ساخت , ZRANGE:خواندن , ZCARD: تعداد عضوها ,

ZCOUNT : تعداد عضوها در رنج , ZINCRBY : اضافه کردن مقداری

ZINTERSTOR : ذخیره کردن مشترک ها , ZPOPMAX , ZPOPMIN ,

ZRANK: چندمین عضو بودن , ZREM:حذف ,

===========================================================

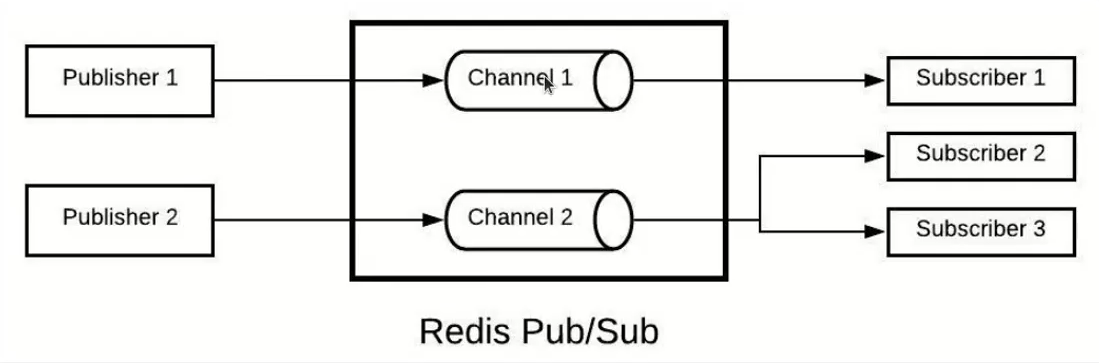
hyperLogLog ها دقیقا مثل set ها هستند یعنی امکان ذخیره موارد تکراری وجود ندارد و اینکه ترتیب ورود اطلاعات مهم نیست. این ساختار برای زمانی مناسب است که خود اطلاعات اهمیتی ندارند و فقط تعداد آنها مهم است. مثلا برنامه ای که به صورت real-time نیاز به ذخیره تعداد کاربرهای unique دارد میتواند از این نوع داده استفاده کند. hyperLogLogها در ردیس بسیار بهینه تر از set ها هستند

دستورات:

PFADD:ساخت , PFCOUNT: تعداد , PFMERGE: ترکیب چندتا و ذخیره

===========================================================

pub/sub یک سیستم پیغام رسانی است که در آن فرستنده (publisher) یک پیغام را در یک کانال(channel) فرستاده و گیرنده(subscriber) با متصل شدن به کانال میتواند پیغام ها را خوانده و پردازش کند. در ردیس گیرنده میتواند به چندین کانال وصل شود و محدودیتی وجود ندارد. در این سیستم فرستنده و گیرنده کاملا از هم جدا(decouple) شده اند. فرستنده ها هیچ اطلاعی از گیرنده ها ندارد و بالعکس گیرنده ها هیچ اطلاعی از فرستنده ها ندارند. این جداسازی باعث شده که سرعت ارسال اطلاعات بیشتر شده و فرستنده منتظر پردازش ها گیرنده نباشد.



دستورات:

SUBSCRIBE: ساخت یک شنونده , PUBLISH: ساخت یک فرستنده

PSUBSCRIBE : گوش دادن به کانالهایی با پترن خاص ,

PUBSUB channels : نمایش کانالها

UNSUBSCRIBE : خارج شدن از کانال

===========================================================

transaction یعنی تراکنش، به مجموعه دستوراتی که در یک صف قرار گرفته و به سرور ردیس ارسال میشوند بک تراکنش گفته میشود. تراکنش ها در redis دو ویژگی اصلی دارند. اول isolated یعنی تمام دستوراتی که در یک transaction قرار میگیرند در یک حالت قرنطینه هستند و اگر کسی خارج از اون transaction اقدام به تغییر اطلاعات کند تمام دستورات abort شده و اجرا نمیشوند. دوم atomic یعنی تمامی دستورات یا باید کاملا اجرا شوند یا اصلا اجرا نشوند، هیچ یک از دستورات نصفه اجرا نخواهد شد.

دستورات:

MULTI: ساخت یک تراکنش , EXEC : اجرای یک تراکنش

DISCARD: بسته یک تراکنشه باز , WATCH: پیداسازی ویژگی ایزوله

===========================================================

تمام تنظیمات سرور ردیس در لینوکس در آدرس /etc/redis و در فایل redis.conf ذخیره می‌شوند

در خود سرور ردیس هم میتوانید با دستور CONFIG GET میتوانید config دستورات را بخوانید و با دستور CONFIG SET میتوانید تغییر دهید:

bind: مشخص میکند چه آی پی هایی میتوانند به سرور شما متصل شوند

requirepass: برای وصل شدن به سرور نیاز به پسورد هست یا نه

maxclients : همزمان چه تعداد کاربر میتواند به سرور متصل شوند

auth : وقتی که پسورد لازم است با این دستور پسورد را وارد کنید

برای بازنویسی فایل config اصلی در ترمینال باید دستور CONFIG REWRITE را بزنید

sudo systemctl start/stop/status/restart redis-server.service

sudo service redis-server start/stop/status/restart

===========================================================

در این قسمت با pipeline در ردیس آشنا میشید. برنامه هایی که با ردیس کار میکنند از طریق یک شبکه به هم متصل شده اند. هر چقدر هم که این شبکه سریع باشه بازهم یک زمان کمی طول میکشه که دستورات از برنامه به ردیس و پاسخ از ردیس به برنامه برسه. به همین خاطر نمیشه از تمام قدرت ردیس استفاده کرد و مثلا چندین هزار داده رو در یک ثانیه ذخیره کرد. برای رفع این مشکل میتوان از pipeline ها استفاده کرد. pipelineها در ردیس به شما اجازه میدن که بتونید چندین درخواست رو داخل یک پکیج یکجا بفرستید و ردیس هم چندین پاسخ رو داخل یک پکیج یکجا برای شما بفرسته.

(prinf “PING\r\nSET name amir\r\nGET name\r\n”;) | nc localhost 6379

همزمان این ۳ دستور را به سرور ردیس ارسال میکند

===========================================================

در این قسمت با مفهوم persistence در ردیس آشنا میشید. همونطور که میدونید ردیس اطلاعات رو داخل ram ذخیره میکنه که سرعت خوندن و نوشتن اطلاعات بالا باشه. حالا فرض کنید که ram ما بسوزه، آیا اطلاعات ما از بین میره؟ قطعا نه، مفهوم persistence میگه که ردیس میتونه به دو روش متفاوت اطلاعات رو داخل Hard بنویسه تا اطلاعات شما در صورت سوختن ram یا ریستارت سرور از بین نره. این دو روش AOF و RDB هستند که در این ویدیو با همه این موارد آشنا میشید.

خودمان میتوانیم با زدن دستور SAVE اطلاعات را ذخیره کنیم

این فایل در لینوکس در آدرس زیر ذخیره می‌شود :

*/var/lib/redis/dump.rdb*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

RDB: بر اساس دیتابیس شما (براساس کاری که دیتابیس شما انجام میدهد) میاد و در زمان های مختلفی اطلاعات دیتابیس شما را داخل فایلی ذخیره میکند

config rdb در redis-cli با دستور CONFIG GET save نمایش داده می‌شود

که خروجی اش به شکل زیر است که مفهومش این است که اگر تغییرات ۱ دونه بود هر ۹۰۰ ثانیه backup بگیر اگر تغییرات 10 تا بود هر 300 ثانیه و اگر 10000 تا تغییر بود هر 60 ثانیه backup بگیر :

900 1 300 10 60 10000

ـــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ

AOF: میاد یک فایل میسازد و همش اطلاعات را به آخر اون فایل میچسباند

ـــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ

اگر بخواهید عملیاتی که با دستور SAVEانجام می‌شود در background برنامه انجام شود میتوانید از دستور BGSAVE استفاده کنیم

ــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ

هنگامی که میخواهید دیتابیس سرور خود را جایگزاری کنید به دلیل اینکه دسترسی سرور از فایل dump از بین رفته است لازم است با زدن دستور زیر دوباره اجازه دسترسی را به سرور ردیس بدهید :

chown redis:redis dump.rdb

توجه شود که باید حتماً در محل اصلی که بالاتر اشاره شد باشید

===========================================================

در این قسمت میبینید که چطور میتونید سرعت سرور ردیس رو تست کنید. برای اینکار ابزاری وجود داره به نامه redis-benchmark که میتونید از اون استفاده کنید. این ابزار زمانی مفید هستش که شما تغییری در ردیس یا برنامتون ایجاد کردین و میخواین ببینین که چه تاثیری روی سرعت گذاشته.

خارج از redis-cli میتوانید دستور زیر را بزنید:

redis-benchmark [option]

===========================================================

در ردیس دیتابیس ها با عدد مشخص می‌شوند

دستورات سروری:

CONFIG GET database : تعداد دیتابیس ها

SELECT 8 : انتخاب دیتا بیس ۸ام

SWAPDB number\_db number\_db : ریختن اطلاعات یک دیتابیس در دیتابیس دیگر

CLIENT LIST : نمایش کاربرانی که به دیتابیس متصل هستند

CLIENT SETNAME : اختصاص نام به کاربر

CLINET GET : نمایش تنها نام کسانی که به دیتابیس متصل شده‌اند

CLIENT KILL : قطع کردن ارتباط کاربری با سرور

FLUSHDB : در دیتابیسی که هستید تمام اطلاعات را حذف میکند

FLUSHALL : حذف تمامی اطلاعات در تمامی دیتابیس ها

===========================================================

برای وصل شدن به ردیس با پایتون میتونید از پکیج redis استفاده کنید. در این پکیج یک کلاس اصلی با نام Redis وجود داره که تمام فانکشن های مربوط به ردیس رو در اختیار شما قرار میده. با استفاده از این پکیج میتونید تمام کارهایی که با redis-cli انجام میدادید رو انجام بدین

در پکیج redis متدی به نام pipeline() وجود داره که به شما یک آبجکت pipeline میده که از اون میتونید برای ساختن یک صف استفاده کنید و دستوراتتون رو در داخل صف قرار بدین و به شکل همزمان به سرور ارسال کنید.