Sử dụng redis-cli

redis-cli là công cụ dòng lệnh có sẵn trong Redis, để kết nối và thực hiện các lệnh Redis (giống mysql trong MySQL Server)

Giả sử để kết nối đến Redis Server ở địa chỉ IP (domain) thì gõ lệnh

redis-cli -h IP

Nếu vào dòng lệnh container đang chạy Redis Server và kết nối đến Redis trong container này thì không cần thiết phải chỉ ra -h IP hoặc muốn thì chỉ ra -h localhost hoặc -h 127.0.0.1

Nếu muốn chỉ ra cổng kết kết thì dùng tham số -p cổng

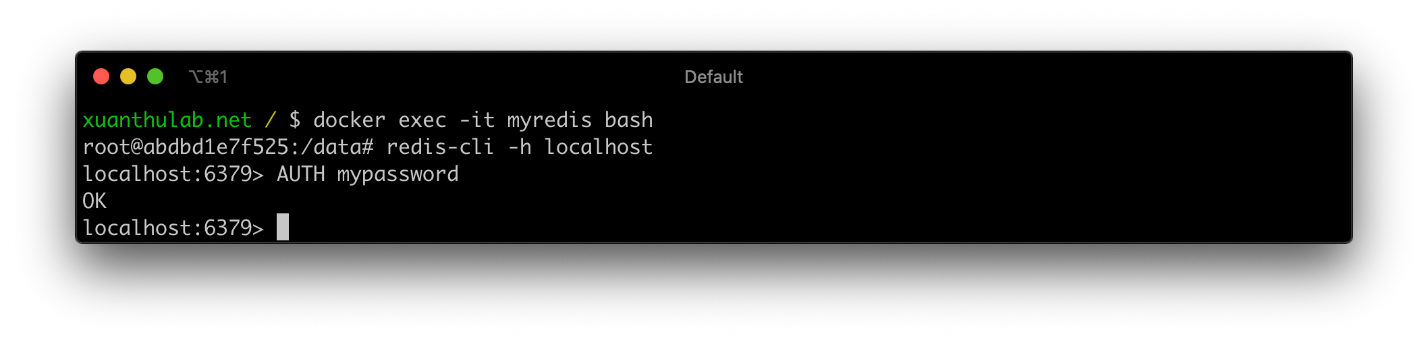
redis-cli -h localhost -p 6391

Sau khi kết nối, thì nó ở dòng lệnh của Redis để thực hiện trực tiếp một số tác vụ mà không cần phụ thuộc ngôn ngữ lập trình nào, một số lệnh có thể tham khảo phần sau:

Một số lệnh Redis

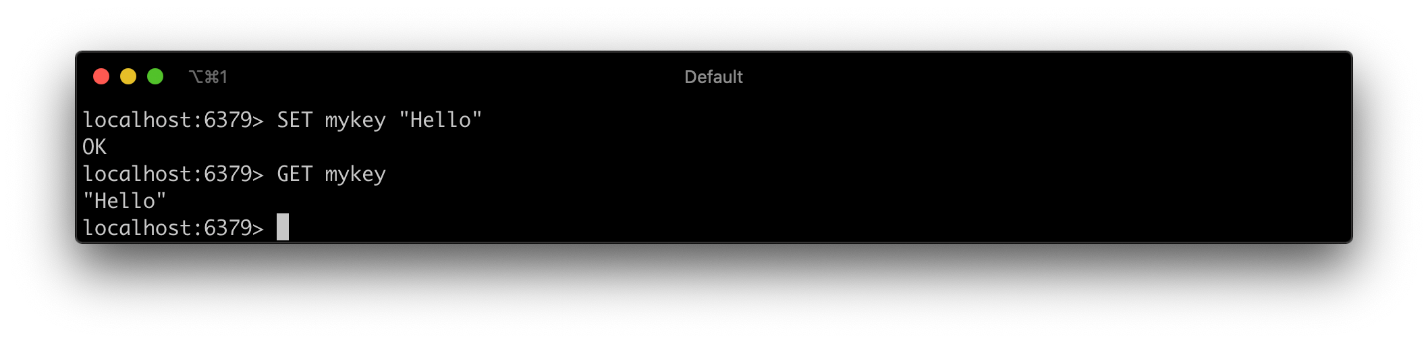
Lệnh AUTH

Để nhập password xác thực, nếu Redis cấu hình cần xác thực



Lệnh SET và GET

Lệnh SET để lưu một giá trị vào một key nào đó, lệnh GET thì để đọc giá trị lưu với key đó



Lệnh DBSIZE

Lệnh DBSIZE để kiểm tra xem có bao nhiêu key dữ liệu đang lưu trữ

Lệnh KEYS

Lệnh này hiện thị tất cả các key

Lệnh DELETE

Lệnh DELETE để xóa một key

DELETE mykey

Lệnh FUSHALL

Xóa tất cả các key trong Redis Server

## Một số kiểu dữ liệu Redis

**String** là kiểu dữ liệu cơ bản nhất của Redis và đây cũng là kiểu dữ liệu duy nhất của Memcache. Có 3 câu lệnh cơ bản với String, đó là GET, SET và DEL.

|  |  |
| --- | --- |
| Câu lệnh | Chức năng |
| GET | Lấy giá trị value của key |
| SET | Thiết lập giá trị cho key |
| DEL | Xóa key và giá trị tương ứng (Làm việc với tất cả kiểu dữ liệu, không chỉ string) |

**List** trong Redis là linked list, lưu trữ 1 danh sách có thứ tự (trước sau) của các string. Cách lưu trữ này giúp cho thời gian add thêm 1 phần tử vào đầu hoặc cuối list là hằng số, bất kể size của list là bao nhiêu. Lợi thế này cũng có 1 mặt trái là việc truy xuất đến phần tử theo index của linked list là lâu hơn rất nhiều so với array.

Dưới đây là các câu lệnh cơ bản làm việc với List

|  |  |
| --- | --- |
| Câu lệnh | Chức năng |
| RPUSH | Cho thêm 1 giá trị vào phía bên phải của list |
| LRANGE | Lấy 1 dải giá trị của list |
| LINDEX | Lấy ra phần tử của list dựa theo index |
| LPOP | Lấy ra giá trị ở phái ngoài cùng bên trái của list và xóa giá trị đó đi |

**Set** trong Redis khá giống với list, nhưng khác 1 điều là các phần tử trong set không được sắp xếp theo thứ tự nào cả. Tuy nhiên, Redis đã tăng performance khi làm việc với set bằng cách sử dụng 1 bảng băm (hash table) để lưu trữ các phần tử của set. Hiểu đơn giản thì mỗi item được add vào set sẽ là 1 key trong bảng băm, còn value thì không có. Việc làm này giúp theo tác truy xuất dữ liệu trên SET nhanh hơn nhiều (do tận dụng ưu thế về tốc độ tìm kiếm trên bảng băm), nhất là khi muốn đảm bảo không bị trùng lặp phần tử trong set.

4 command cơ bản khi làm việc với set là SADD, SMEMBERS, SISMEMBER, SREM. Chức năng có thể tham khảo bảng sau.

|  |  |
| --- | --- |
| Câu lệnh | Chức năng |
| SADD | Add thêm 1 phần tử vào set |
| SMEMBERS | Lấy ra tất cả các phần tử của set |
| SISMEMBER | Check xem 1 phần tử có tồn tại trong set hay không |
| SREM | Xóa đi 1 phần tử của set (nếu nó tồn tại) |

**HASH** Không giống như LIST và SET lưu trữ 1 tập dữ liệu là các string, HASH lưu trữ tập các map của key và value. Key vẫn là string, còn value có thể là string hoặc số. Nếu là số thì chúng ta có thể làm các thao tác tăng, giảm giá trị 1 cách đơn giản. HASH được coi là 1 mô hình thu nhỏ của Redis, khi dữ liệu được tổ chức dạng key-value. Bảng sau liệt kê các câu lệnh cơ bản khi làm việc với HASH

|  |  |
| --- | --- |
| Câu lệnh | Chức năng |
| HSET | Thêm 1 cặp key-value vào hash hoặc thay đổi giá trị của key đã có. |
| HGET | Lấy giá trị của key trong hash có sẵn |
| HGETALL | Lấy ra tất cả các phần tử của hash |
| LDEL | Xóa cặp key-value ra khỏi hash (nếu key tồn tại) |