

Tham Khảo Nhanh Redis

String, list, set, hash, pub/sub và tính bền vững

Kết Nối

CLI

```
redis-cli
redis-cli -h 127.0.0.1 -p 6379
redis-cli -a password -n 2
redis-cli --tls -u rediss://user:pass@host:6380
```

Kết Nối Driver (Python)

```
import redis
r = redis.Redis(host='localhost', port=6379, db=0)
r.set('key', 'value')
print(r.get('key'))
```

Thông Tin Server

```
PING -- returns PONG
INFO server -- server details
INFO memory -- memory usage
DBSIZE -- number of keys in current db
```

String

Thao Tác Cơ Bản

```
SET name "Alice"
GET name
SET counter 100
MSET a 1 b 2 c 3
MGET a b c
```

Thao Tác Số

```
INCR counter -- 101
INCRBY counter 10 -- 111
DECR counter -- 110
DECRBY counter 5 -- 105
INCRBYFLOAT price 2.5
```

Lệnh String

SET key val	Gán giá trị string
GET key	Lấy giá trị string
SETNX key val	Gán chỉ khi key chưa tồn tại
SETEX key sec val	Gán với thời gian hết hạn (giây)
APPEND key val	Nối thêm vào giá trị hiện tại
STRLEN key	Độ dài giá trị string

List

Thao Tác List

```
LPUSH queue "first"
RPUSH queue "last"
LRANGE queue 0 -1 -- all elements
LPOP queue
RPOP queue
```

Lệnh List

LPUSH / RPUSH	Đẩy vào đầu / cuối list
LPOP / RPOP	Lấy từ đầu / cuối list
LRANGE key start stop	Lấy dài phần tử
LLEN key	Độ dài list
LINDEX key idx	Phần tử tại vị trí
LREM key count val	Xóa count lần xuất hiện của val
BLPOP key timeout	Lấy chặn từ đầu list (dùng cho queue)

Set & Sorted Set

Thao Tác Set

```
SADD tags "python" "redis" "docker"
SMEMBERS tags
SISMEMBER tags "python" -- 1 (true)
SREM tags "docker"
SCARD tags -- count
```

Phép Toán Tập Hợp

```
SUNION set1 set2 -- union
SINTER set1 set2 -- intersection
SDIFF set1 set2 -- difference
```

Thao Tác Sorted Set

```
ZADD leaderboard 100 "Alice" 85 "Bob"
ZRANGE leaderboard 0 -1 WITHSCORES
ZREVRANGE leaderboard 0 2
ZSCORE leaderboard "Alice"
ZRANK leaderboard "Alice" -- 0-based rank
```

Lệnh Sorted Set

ZADD key score member	Thêm thành viên với điểm số
ZRANGE key start stop	Dài theo thứ hạng (thấp đến cao)
ZREVRANGE key start stop	Dài theo thứ hạng (cao đến thấp)
ZINCRBY key incr member	Tăng điểm số thành viên
ZRANGEBYSCORE key min max	Dài theo giá trị điểm số
ZCARD key	Số lượng thành viên

Hash

Thao Tác Hash

```
HSET user:1 name "Alice" age 30
HGET user:1 name
HGETALL user:1
HMSET user:2 name "Bob" age 25
HMGET user:1 name age
```

Lệnh Hash

HSET key field val	Gán trường hash
HGET key field	Lấy trường hash
HGETALL key	Lấy tất cả trường và giá trị
HDEL key field	Xóa trường hash
HEXISTS key field	Kiểm tra tồn tại trường
HINCRBY key field n	Tăng giá trị trường
HKEYS key	Tất cả tên trường
HLEN key	Số lượng trường

Key & Hết Hạn

Lệnh Key

KEYS pattern	Tìm key khớp mẫu (chậm)
SCAN cursor MATCH pat	Duyệt key dần dần (an toàn)
EXISTS key	Kiểm tra key tồn tại
DEL key	Xóa key
TYPE key	Lấy kiểu dữ liệu của key
RENAME key newkey	Đổi tên key

Lệnh Hết Hạn

```
EXPIRE key 3600 -- expire in 1 hour
PEXPIRE key 5000 -- expire in 5000 ms
TTL key -- seconds until expiry
PTTL key -- ms until expiry
PERSIST key -- remove expiry
```

Mẫu Key

```
SET session:abc123 "data" EX 1800
-- EX = seconds, PX = milliseconds
-- NX = only if not exists
-- XX = only if exists
SET lock:order42 "owner" NX EX 10
```

Pub/Sub

Pub/Sub Cơ Bản

```
-- Subscriber (terminal 1)
SUBSCRIBE news alerts

-- Publisher (terminal 2)
PUBLISH news "Breaking: Redis 8 released"
```

Subscribe Theo Mẫu

```
PSUBSCRIBE news.*
-- matches news.tech, news.sports, etc.
```

Lệnh Pub/Sub

SUBSCRIBE channel	Lắng nghe tin nhắn trên kênh
PUBLISH channel msg	Gửi tin nhắn tới kênh
PSUBSCRIBE pattern	Subscribe theo mẫu
UNSUBSCRIBE channel	Dừng lắng nghe
PUBSUB CHANNELS	Liệt kê kênh đang hoạt động

Transaction

MULTI / EXEC

```
MULTI
SET balance:1 900
SET balance:2 1100
EXEC -- executes atomically
```

Khóa Lạc Quan

```
WATCH balance:1
val = GET balance:1 -- read current
MULTI
SET balance:1 (val - 100)
EXEC
-- EXEC returns nil if balance:1 changed
```

Lệnh Transaction

MULTI	Bắt đầu khối transaction
EXEC	Thực thi các lệnh đã xếp hàng
DISCARD	Hủy các lệnh đã xếp hàng
WATCH key	Theo dõi key thay đổi (khóa lạc quan)
UNWATCH	Bỏ theo dõi tất cả key

Tính Bền Vững

Snapshot RDB

```
SAVE -- synchronous snapshot
BGSAVE -- background snapshot
LASTSAVE -- timestamp of last save
```

AOF (Append Only File)

appendonly yes	Bật AOF trong redis.conf
appendfsync always	Fsync mỗi lần ghi (an toàn nhất, chậm nhất)
appendfsync everysec	Fsync mỗi giây (khuyến nghị)
appendfsync no	Để OS quyết định (nhanh nhất, rủi ro nhất)

Tham Khảo Nhanh Redis

Lệnh Tính Bền Vững

```
CONFIG GET save
CONFIG SET save "900 1 300 10"
-- snapshot if 1 change in 900s or 10 in 300s
BGREWRITEAOF -- rewrite AOF in background
```

Mẫu Phổ Biến

Khóa Phân Tán

```
SET lock:resource "owner-id" NX EX 30
-- NX = acquire only if not held
-- EX 30 = auto-release after 30s
DEL lock:resource -- explicit release
```

Giới Hạn Tốc Độ

```
key = "rate:user:42"
INCR key
EXPIRE key 60 -- 60-second window
-- reject if GET key > max_requests
```

Mẫu Cache

```
val = GET "cache:user:1"
if val is nil:
  val = fetch_from_db(1)
  SET "cache:user:1" val EX 300
```

Lưu Trữ Session

```
HSET sess:abc uid 42 role "admin"
EXPIRE sess:abc 1800 -- 30 min TTL
HGETALL sess:abc
```