

# 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

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

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

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

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

## Key & Hết Hạn

### Lệnh Key

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

<b>SUBSCRIBE channel</b>	Lắng nghe tin nhắn trên kênh
<b>PUBLISH channel msg</b>	Gửi tin nhắn tới kênh
<b>PSUBSCRIBE pattern</b>	Subscribe theo mẫu
<b>UNSUBSCRIBE channel</b>	Dừng lắng nghe
<b>PUBSUB CHANNELS</b>	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

<b>MULTI</b>	Bắt đầu khối transaction
<b>EXEC</b>	Thực thi các lệnh đã xếp hàng
<b>DISCARD</b>	Hủy các lệnh đã xếp hàng
<b>WATCH key</b>	Theo dõi key thay đổi (khóa lạc quan)
<b>UNWATCH</b>	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)

<b>appendonly yes</b>	Bật AOF trong redis.conf
<b>appendfsync always</b>	Fsync mỗi lần ghi (an toàn nhất, chậm nhất)
<b>appendfsync everysec</b>	Fsync mỗi giây (khuyến nghị)
<b>appendfsync no</b>	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
```