# 阿里云 云数据库 Redis 版

20190214



# <u>法律声明</u>

- 1.
- 2.
- 3.
- 4. " " "

- 5. / /
- " "Aliyun" "
- " / /
- 6.

# 通用约定

•		
A		▲ : 10
		: Ctrl + A
>		> >
	UI	
courier		cd /d C:/windows Windows
##		bae log listinstanceid Instance_ID
[] [a b ]		ipconfig [-all -t]
{} {a b }		<pre>swich {stand   slave}</pre>

# 目录

		 	 I
		 	 I
1	Redis	 	 
2	Redis	 	 2
3		 	 7
4 Redis		 	 
5		 	 15
		 	 17
7		 	 
7.1 DN	//S	 	 
7.2 Re	dis	 	 
7.3 re	dis-cli	 	 31
7.4		 	 

# 1开始使用云数据库Redis版

#### 文档目的



# 2 云数据库Redis版管理控制台

Red	lis	Redis	Web	
Rec	lis	Redis		
前提条件				
		Redis		
控制台简	介			
	Redis			

### Redis

云数据库Redis版	实例列表 华北1 华北2 华东	1 华东 2 华南 1 香港 亚太东	3比 1 (东京) 亚太东南	1 (新加坡)	亚太东南 2 (悉尼) — 剪	美国东部 1 (弗吉尼亚) 美国西部	第1(硅谷)		います	创建实例
实例列表	BC用中部 1 (法三元福)									
	実例ID ▼ 清输入完整实例ID,多个II	D使用","分隔 投索								
	其例ID/名称 状态	已用内存及配额	可用区	实例规格	创建时间	付费方式	网络类型	2017年5月SLA		操作
	r-bp1352718e327064 256test 使用中	291.16MB/64.00GB(0.44%)	华东 1 可用区 E	64G爆散版	2017-05-12 10:54	包年包月 到期时间:2017-06-13	经典网络	计脚中	管理 续	想 交更配置
	□ r-bp1353509ae80ca4 使用中	33.03MB/1.00GB(3.23%)	华东 1 可用区 E	1G主从版	2016-12-19 17:53	包年包月 到期时间:2017-03-11	经典网络	计算中	管理 续	费 交更配置
	日 社最续要 <b>申请昭付</b>	申请复议					共有2条	,每页显示:20条	« < 1	5

ID

(!) :		
		10
DM	1S D!	MS

Redis

<	😝 256test		發型数据库
突例信息 性能监控	实例信息		
报警设置	基本信息		~
参数设置	实例ID:	实例名称: 🐜 🖬 🖌	状态: 使用中
昏份与恢复	可用区: 华东 1 可用区 E	网络英型: 经 <b>典网络</b>	1984/191000           © 22:00 02:00         © 02:00 06:00           © 66:00 10:00         © 10:00-14:00           © 14:00-18:00         © 18:00-22:00           %7 10:75         %7
	实例规格: 64G集群版	最大连接数: 80000	最大内网带宽: 384 MByte

# 性能监控

١D

Redis

<	256best	登录数据率 迁移数据率 C 副新 修改能药 清除数据 Q 型配器
2012B	历史监控数据	
RURE	thilting Kop Expires ExpiredKeys ExcitedKeys UsedMemory Inflow OutFlow FailedCount	ConnCount TotalQos Cpubliage +40093281938
学校设置	#HittensHi : 2017-06-02 15:00 - 2017-06-02 16:00	
=		
	0	
		8
	15.00 15.05.00 15.15.00 15.15.00 15.20.00 15.25.00 15.30.00	15:35:00 15:40:00 15:45:00 15:50:00 15:56:00 16:0

Keys	Redis db key	
Expires	key	
ExpiredKeys	key key key :	key
EvictedKeys	key	key key
UsedMemory	30MB 200MB	30MB

InFlow	Redis			K B	ytes/s			
OutFlow	Redis		K Bytes/s					
FailedCount			Proxy Faile	Redis dCount		Redis	DB	
ConnCount	Redis							
TotalQps	Redis							
CpuUsage	Redis	CPU						

## 

set

# 报警设置

<	256test			建异物编样	TRAILS	C RIT	修改密码	清涉政报	\$28A
实例信息	报警设置								800X
世紀高校									
RURE	121278	[6789-620]	86117888	秋古	.010		REDAX		
学致设置									
备份均依额			① 没有查询到符合	除件的记录					
	注:服整合物转到云监控设置								

Redis

Redis

78篇:	云数据库Reds版	٣									
资源范围:	全部资源	• 0									
设置报警规则											
报警供型:	國協設部 事件报警										
规则名称:	内存报警					80.00					
-	CONTRACT OF			-		60.00					
MONTHINE :	PHYRHA	* 300 NP	• <b>*</b> ***	• • •	80 76	40.00					
+/#2018	督规则					20.00					
连续几次超3 傅值后报警	1 .					0.74					
主动时间:	00:00 * 至 23:59	*				17:10:00	11:0 ● 内北	他用老 <u>一</u> 平均6	100 14	<ul> <li>53:20 04</li> <li>53:20 04</li> </ul>	90090 17:00
						波穿范围选择"应用分 新会和山田町通20		的"全部资源"》	1,分组内成产	品内任何资源》	
通知方式 —						资源范围选择"应用分 都会发出报警遭知。 折线图展示分组内部	∃组"成≃后 分实例的 <sup>2</sup>	的"全部资源"的	1,分组内成本 ,为您设置报警	品内任何资源》	
通知方式 — 通知对象:	10.86人通知道	<u>28</u>	E	己选进1个	<u>28</u>	资源范围选择"应用分 都会发出报警通知。 所线图最示分组内部	∃狙"或产品 分实例的 <sup>2</sup>	的"全部资源") "均聚合值主势	1,分组内成产 ,为您设置限制	品内任何旅源源 國內任何旅源源	
遵知方式 通知对象:	N-8人唐10日 夜歌	<u>≑të</u> Q	E	已透過1个 云外号报警教系人	<del>23</del>	资源范围选择"应用分 都会发出报警遇知。 拆线图最示分组内部	3组"成产品 分实例的 <sup>3</sup>	的"全部资源"。 "均聚合值主势	1,分组内成24	品内任何波察課 調査提供参考。	12/22/2000
遵知方式 — 通知对象:	RES.人通知28 序定 GPU高控	see Q	4	□遗辑 1 个 云师号报督联系人	źż	资源范围选择" 拉用 分 都会发进限警遇知。 所线图展示分组内部	381 "或产品 (分实例的)"	的"全部资源"。	1,分组内成24	国内任何批算課	
遵知方式 — 通知对象:	R系人語0回 原定 GPU直12	<u>\$8</u> Q	•	已感道 1 个 云师母姐驾驶系人	<u>28</u>	推荐范围选择"拉局分 都会发出预算通知。 所线图带示分组内部	分组"规产品 分类例的 <sup>34</sup>	的"全部资源"日 "均聚合值主势	1,分组内或24	■ 小田石(田- 品内任何流源課 調査提供参考。	
遵知方式 一 通知对象:	取高人通知0回 15日 GPU国政	<del>≩tă</del> Q	+	二歲頃1个 云称号張醫教集人	<u>\$8</u>	资源范围选择"应用分 都会发出场智通知。 所线图积示分组内部	分支例的不	品?"金師或譯"B	1 , 分組内認2 <sup>4</sup>	→ 外部為《Ш 品内任何流源課 時後證供参考,	4.2.按整規則能力
遵知方式	取高人通い回 示示 GPU国社 快速的開取高人団	Q Q	•	三歲頃 1 个 云班母留智敏美人	÷ž	追摩范重选择"应用分 都会及过时智速见。 所线图展示分组内部	分实例的	的"全部遗源"口	1 , 分組内網24	■ 20004(m 品内任何波源課 詞面環代参考。	4.2 按整项印版社
遵知方式 一 违知对象: 通知方式:	取用 - 2015年 のの の 取用 - 2015年 の の に 入 の の の に 入 の の の の の の の の の の の の の	28 Q	+	3.微磁 1 个 云称号倍器联系人	<u>\$8</u>	進厚亞憲法律" (4月15年 基本2011年間道知1。 所线图展示分组内部	→服"呢2本品 分类例的3	的"全部强厚"。 "均聚合值走势	1,分组内和24	● 外部なく(m 品内任何道理時 時後道理(1参考)	4.2 按整规则重社
遵知方式 — 通知对象: 通知方式: 邮件编注:	03.7.8502 572 GYU232 GUIDED 5.7.02 BUIDED 5.	23 Q -	•	3.端道 1 个 云称号倍智教素人	22	進費原面佔非 自用品 書完大出用管理机 所任意展示分组內部	→銀 <sup>™</sup> 成本品 分支表的内	的"全部资源"口	1, 39309982	● 外部石(山 品内任何法理時 総裁證(1参考)	42:按器规 P3至6
護知方式 速和功象: 速和方式: 邮件编注:	0.05.4800 877 07022 0020000.5.40 002000.5.40 884-822-00708A	28 Q *	+	2000년 1 个 云称号级智敏素人	ŝŝ	此原作面也详" 如此中 都方式出版智速化。 所试图展示分组内部	"你不知道""这个最	的"全部资源"	1, 593879857	● 外部なく(m 品内任何法理時 時後部連代参考。	

### 参数设置

Redisnotify-keypsace-events备份恢复

## DMS页面

DMS

Key

# ExpiredKeys EvictedKeys



List	List 8192 value	512 MB 1 MB	list
Set	set 8192 value	512 MB 1 MB	set
Sorted set	sorted set set 8192 val	512 ue 1 M	MB sorted 1B
Hash	field 8192 value	512 MB 1 MB	
DB	256 DB		
	DB DB dbO	db	
Redis	Redis		
	Redis		
	•		key
	•	key key	
	Redis		
	AOF_FSYNC_EVERYSE	C fysnc	

# 4 Redis命令

Redis	2.8	4.0	Redis		Redis
Redis	Redis	3.0	Redis 3.0	Geo	

# 支持的Redis命令

### 表 4-1: Redis命令支持表1

Keys	String	Hash	List	Set	SortedSet
DEL	APPEND	HDEL	BLPOP	SADD	ZADD
DUMP	BITCOUNT	HEXISTS	BRPOP	SCARD	ZCARD
EXISTS	BITOP	HGET	BRPOPLPUSH	SDIFF	ZCOUNT
EXPIRE	BITPOS	HGETALL	LINDEX	SDIFFSTORE	ZINCRBY
EXPIREAT	DECR	HINCRBY	LINSERT	SINTER	ZRANGE
MOVE	DECRBY	HINCRBYFLO AT	LLEN	SINTERSTOR E	ZRANGEBYSC ORE
PERSIST	GET	HKEYS	LPOP	SISMEMBER	ZRANK
PEXPIRE	GETBIT	HLEN	LPUSH	SMEMBERS	ZREM
PEXPTREAT	GETRANGE	HMGET	LPUSHX	SMOVE	ZREMRANGE YRANK
PTTL	GETSET	HMSET	LRANGE	SPOP	ZREMRANGEE YSCORE
RANDOMKEY	INCR	HSET	LREM	SRANDMEMB R	EREVRANGE
RENAME	INCRBY	HSETNX	LSET	SREM	ZREVRANGEB YSCORE
RENAMENX	INCRBYFLOA T	HVALS	LTRIM	SUNION	ZREVRANK
RESTORE	MGET	HSCAN	RPOP	SUNIONSTOR E	ZSCORE
SORT	MSET		RPOPLPUSH	SSCAN	ZUNIONSTOR E
TTL	MSETNX		RPUSH		ZINTERSTOR E

Keys	String	Hash	List	Set	SortedSet
ΤΥΡΕ	PSETEX		R P U SH X		ZSCAN
SCAN	SET				ZRANGEBYLE X
OBJECT	SETBIT				ZLEXCOUNT
	SETEX				ZREMRANGEB YLEX
	SETNX				
	SETRANGE				
	STRLEN				

# 表 4-2: Redis命令支持表2

HyperLogLo g	Pub/Sub /	Transactio n	Connection	Server	Scripting	Geo
PFADD	PSUBSCRIB	EDISCARD	AUTH	FLUSHALL	EVAL	GEOADD
PFCOUNT	PUBLISH	EXEC	ЕСНО	FLUSHDB	EVALSHA	GEOHASH
PFMERGE	PUBSUB	MULTI	PING	DBSIZE	SCRIPT EXISTS	GEOPOS
	PUNSUBSCI BE	RWNWATCH	QUIT	TIME	SCRIPT FLUSH	GEODIST
	SUBSCRIBE	WATCH	SELECT	INFO	SCRIPT KILL	GEORADIUS
	UNSUBSCR E	В		KEYS	SCRIPT LOAD	GEORADIUSE YMEMBER
				CLIENT KILL		
				CLIENT LIST		
				CLIENT GETNAME		
				CLIENT SETNAME		

HyperLogLo	Pub/Sub	Transactio	Connection	Server	Scripting	Geo
g	/	n				
				CONFIG		
				GET		
				MONITOR		
				SLOWLOG		



# 

- Redis clientlist proxy user connection id age idle addr fd name db multi omem cmd Redis sub psub proxy 1 O qbuf qbuf-free obl oll
- Redis clientkill client kill ip:port client kill addr ip:port

### 4.0版本新增Redis命令

#### 表 4-3: 4.0 Redis命令

Keys	Server
UNLINK	SWAPDB
	MEMORY

4.0 Redis

FLUSHALL/FLUSHDB ASYNC

ASYNC FLUSHALL FLUSHDB Redis 4.0 <i>Redis 4.0</i>			
Redis 4.0 <i>Redis 4.0</i>	ASYNC	FLUSHALL FLUSHDB	
	Redis 4.0	Redis 4.0	

### 暂未开放的Redis命令

Keys	Server
MIGRATE	BGREWRITEAOF
	BGSAVE

Keys	Server
	CONFIG REWRITE
	CONFIG SET
	CONFIG RESETSTAT
	COMMAND
	COMMAND COUNT
	COMMAND GETKEYS
	COMMAND INFO
	DEBUG OBJECT
	DEBUG SEGFAULT
	LASTSAVE
	ROLE
	SAVE
	SHUTDOWN
	SLAVEOF
	SYNC

# 集群实例受限制的Redis命令

Keys	Strings	Lists	HyperLogLo	Transaction	Scripting
			g		
RENAME	MSETNX	RPOPLPUSH	PFMERGE	DISCARD	EVAL
RENAMENX		BRPOP	PFCOUNT	EXEC	EVALSHA
SORT		BLPOP		MULTI	SCRIPT EXISTS
		BRPOPLPUSH		UNWATCH	SCRIPT FLUSH
				WATCH	SCRIPT KILL
					SCRIPT LOAD



:

•	Redis		key	hash slot		
	hash slot		hash tag	k	tey	hash
	slot					
	key1 aakey a	abkey3				
	{key}1 aa{key} ab	o{key}3				hash
	tag Redis		http://redis.io/topics/clu	ster-spec		
•	watch	h	key		key	
	slot	redis			key	
	slot					

key DEL SORT MGET MSET BITOP EXISTS MSETNX RENAME RENAMENX BLPOP BRPOP RPOPLPUSH BRPOPLPUSH SMOVE SUNION SINTER SDIFF SUNIONSTORE SINTERSTORE SDIFFSTORE ZUNIONSTORE ZINTERSTORE PFMERGE PFCOUNT WATCH UNWATCH RANDOMKEY KEYS UNSUBSCRIBE PSUBSCRIBE PUNSUBSCRIBE PUBLISH SUBSCRIBE PUBSUB SCRIPT EVAL EVALSHA SCAN ISCAN DBSIZE ADMINAUTH AUTH PING ECHO FLUSHDB FLUSHALL MONITOR IMONITOR RIMONITOR INFO IINFO RIINFO CONFIG SLOWLOG TIME CLIENT

Lua

Lua -

- key KEYS redis.call/pcall redis key KEYS "-ERR bad lua KEYSarray Lua script for redis cluster, all the keys that the script uses should be passed using the KEYS array\r\n"
- key 1 slot "-ERR eval/evalsha command keys must be in same slot\r\n"
- "-ERR for redis cluster, eval/ key evalsha number of keys can't be negative or zero\r\n"

#### 自研的集群实例Redis命令

 infokey key slot db Redis info Redis section (info [section]) key

	slot	info key	key	slot db	
	127.0.0.1: slot:15118	5379> info key te node_index:0	est_key		
	()				
	-	info key	node index	node -	index
	- info key	node		select db	
•	iinfo	info	Redis ir	ıfo	
	iinfo db_idx	[section]			
	db_idx info	[O, nodecount] section	nodecount Redis inf <i>Redis</i>	info sec To iinfo	tion
•	riinfo iir	fo			readonly
	slave idx readonlys	readon slave info	ly slave info		
	riinfo db_id	<pre> ro_slave_idx [s</pre>	section]		
•	iscan	db_idx db_idx	db scan [O, nodecount]	scan nodecount	info
	iscan db_idx	cursor [MATCH pa	attern] [COUNT cou	unt]	
•	imonitor db_idx db_idx	iinfo iscan [O, nodecou	monitor int) nodecount	m oi info	nitor

imonitor db\_idx

•	rimonitor monitor	riinfo			shard	
	rimonito	r db_idx ro <u></u>	_slave_idx			
	im on i tor	: rimonitor	telnet	imoni	tor/rimonitor	quit
说明	5 11					
•	Redis			unkown con	nmand	

• Redis

# 5 创建实例



32 M B

\*32 MB

Redis

4

https://cloud.video.taobao.com/play/u/3050941791/p/1/e/6/t/1/56539900.mp4

# 相关API

CreateInstance

IΡ

# 6设置IP白名单

背景信息

Redis	IP	

Redis

前提条件



### 操作步骤





•

•		Redis	IP	IP	IP	0.0.0/0
		127.0.0	.1			
-	IP	10.10.10.0/24		10. 10. 10. X	IP	Redis
-		IP				
-			1000 II	P		

# 7 连接实例

# 7.1 DMS登录云数据库

[	DMS Da		Web			
Redis	MySQL	SQL Server	PostgreSQL	MongoDB	DMS	

DMS



从Redis管理控制台登录

1.		Redis							
2.									
3.							ID		
4.									
			:						
	•		DMS	Redis		DMS	IP	Redis	
		DMS	<b>⑥务器尚未</b>	加入到RDS白名单	L				$\times$
			) 当前实 单:1	9例尚未设置DMS白名 00.104.175.0/24	3.单导致访问失	:败,需【实	例购买者】添加	对应DMS服务器	器白名
									关闭
	•		D	MS					
	•				VPC	<u>)</u>			

1. *DMS* 

2.

	■ RDS数据库登录 <sup>動業→</sup>								
	redis.rds.a	aliyuncs.com:6379			~				
	•••••								
	☑ 记住密码								
		登	큓						
	关于DMS(Data Management Service) Copyright © DMS All Rights Reserved (Alibaba 数据管理产品)								
•	Redis								
•	VPC	DMS	DMS	VPC					

DMS *DMS* 

# 7.2 Redis客户端连接



			/
Redis			
Redis	FCS	Redis	

Redis

- Jedis
- phpredis
- redis-py
- *C*/*C*++
- .*net*
- node-redis
- C# StackExchange.Redis

http://redis.io/clients

#### Jedis客户端

Jedis Redis

- Jedis
- JedisPool

#### 1. Jedis

2. Jedis

```
a. Eclipse Project
```

```
import redis.clients.jedis.Jedis;
public class jedistest {
public static void main(String[] args) {
try {
      String host = "xx.kvstore.aliyuncs.com";//
      int port = 6379;
      Jedis jedis = new Jedis(host, port);
      11
      jedis.auth("password");//password
String key = "redis";
      String value = "aliyun-redis";
      //select db
                         0
      jedis.select(1);
      //set
                key
      jedis.set(key, value);
      System.out.println("Set Key " + key + " Value: " + value);
      //get
                       key
     String getvalue = jedis.get(key);
System.out.println("Get Key " + key + " ReturnValue: " +
getvalue);
      jedis.quit();
      jedis.close();
}
catch (Exception e) {
```

```
e.printStackTrace();
       }
      }
      Ĵ,
  b.
            Project
                      Eclipse
    Redis
      Set Key redis Value aliyun-redis
      Get Key redis ReturnValue aliyun-redis
                                    Jedis
                                                        Redis
    JedisPool
                                    Redis
3. JedisPool
                              Project
  а.
        Eclipse
                                          pom
      <dependency>
      <proupId>redis.clients</proupId>
      <artifactId>jedis</artifactId>
      <version>2.7.2</version>
      <type>jar</type>
      <scope>compile</scope>
      </dependency>
  b. project
      import org.apache.commons.pool2.PooledObject;
      import org.apache.commons.pool2.PooledObjectFactory;
      import org.apache.commons.pool2.impl.DefaultPooledObject;
      import org.apache.commons.pool2.impl.GenericObjectPoolConfig;
      import redis.clients.jedis.HostAndPort;
      import redis.clients.jedis.Jedis;
      import redis.clients.jedis.JedisPool;
      import redis.clients.jedis.JedisPoolConfig;
            Jedis
                            Jedis-2.7.2
                                         Project
  C.
      JedisPoolConfig config = new JedisPoolConfig();
                                            ApsaraDB for Redis
      //
      config.setMaxIdle(200);
                                        ApsaraDB for Redis
      //
      config.setMaxTotal(300);
      config.setTestOnBorrow(false);
      config.setTestOnReturn(false);
      String host = "*.aliyuncs.com";
      String password = "
                              JedisPool pool = new JedisPool(config, host, 6379, 3000, password
      );
      Jedis jedis = null;
      try {
      jedis = pool.getResource();
      /// ... do stuff here ... for example
     jedis.set("foo", "bar");
String foobar = jedis.get("foo");
jedis.zadd("sose", 0, "car");
jedis.zadd("sose", 0, "bike");
```

```
Set<String> sose = jedis.zrange("sose", 0, -1);
   } finally {
   if (jedis != null) {
   jedis.close();
    /// ... when closing your application:
   pool.destroy();
                            Jedis-2.6 Jedis-2.5
d.
          Jedis
                                                    Project
   JedisPoolConfig config = new JedisPoolConfig();
                                              ApsaraDB for Redis
   //
   config.setMaxIdle(200);
                                         ApsaraDB for Redis
    //
   config.setMaxTotal(300);
   config.setTestOnBorrow(false);
   config.setTestOnReturn(false);
   String host = "*.aliyuncs.com";
   String password = "
                              . . . .
   JedisPool pool = new JedisPool(config, host, 6379, 3000, password
    );
    Jedis jedis = null;
   boolean broken = false;
    try {
          jedis = pool.getResource();
          /// ... do stuff here ... for example
         /// ... do sturr here ... for example
jedis.set("foo", "bar");
String foobar = jedis.get("foo");
jedis.zadd("sose", 0, "car");
jedis.zadd("sose", 0, "bike");
Set<String> sose = jedis.zrange("sose", 0, -1);
   }
   catch(Exception e)
    Ł
          broken = true;
    } finally {
   if (broken) {
          pool.returnBrokenResource(jedis);
   } else if (jedis != null) {
          pool.returnResource(jedis);
   }
}
```

e. Project Eclipse

Redis

Set Key redis Value aliyun-redis Get Key redis ReturnValue aliyun-redis

Jedis

Redis

phpredis客户端

1. phpredis

2.

php

```
<?php
                     host port */
 /*
$host = "localhost";
port = 6379;
                       password */
 /*
                id
$user = "test_username";
$pwd = "test_password";
$redis = new Redis();
if ($redis->connect($host, $port) == false) {
         die($redis->getLastError());
  }
if ($redis->auth($pwd) == false) {
         die($redis->getLastError());
 }
 /*
                                       https://github.com/phpredis/
phpredis */
if ($redis->set("foo", "bar") == false) {
         die($redis->getLastError());
$value = $redis->get("foo");
echo $value;
?>
                                                        Redis
                                   phpredis
```

https://github.com/phpredis/phpredis

#### redis-py客户端

З.

1.	redis-py
2.	Python

```
C/C++客户端
```



git clone https://github.com/redis/hiredis.git

redis-py

2.

```
cd hiredis
     make
     sudo make install
C/C++
     #include <stdio.h>
     #include <stdlib.h>
     #include <string.h>
     #include <hiredis.h>
     int main(int argc, char **argv) {
     unsigned int j;
     redisContext *c;
     redisReply *reply;
     if (argc < 4) {
              printf("Usage: example xxx.kvstore.aliyuncs.com 6379
instance_id password\n");
              exit(0);
     }
     const char *hostname = argv[1];
     const int port = atoi(argv[2])
     const char *instance_id = argv[3];
const char *password = argv[4];
     struct timeval timeout = \overline{\{1, 500000\}}; // 1.5 seconds
     c = redisConnectWithTimeout(hostname, port, timeout);
if (c == NULL || c->err) {
     if (c) {
              printf("Connection error: %s\n", c->errstr);
              redisFree(c);
     } else {
              printf("Connection error: can't allocate redis context\
n");
     }
     exit(1);
     ł
     /* AUTH */
     reply = redisCommand(c, "AUTH %s", password);
     printf("AUTH: %s\n", reply->str);
     freeReplyObject(reply);
     /* PING server */
     reply = redisCommand(c,"PING");
     printf("PING: %s\n", reply->str);
     freeReplyObject(reply);
     /* Set a key */
     reply = redisCommand(c,"SET %s %s", "foo", "hello world");
     printf("SET: %s\n", reply->str);
     freeReplyObject(reply);
     /* Set a key using binary safe API */
     reply = redisCommand(c,"SET %b %b", "bar", (size_t) 3, "hello",
 (size_t) 5);
     printf("SET (binary API): %s\n", reply->str);
     freeReplyObject(reply);
     /* Try a GET and two INCR */
     reply = redisCommand(c,"GET foo");
     printf("GET foo: %s\n", reply->str);
     freeReplyObject(reply);
     reply = redisCommand(c,"INCR counter");
     printf("INCR counter: %lld\n", reply->integer);
     freeReplyObject(reply);
     /* again ... */
     reply = redisCommand(c,"INCR counter");
     printf("INCR counter: %lld\n", reply->integer);
```

freeReplyObject(reply);

7

```
/* Create a list of numbers, from 0 to 9 */
reply = redisCommand(c,"DEL mylist");
freeReplyObject(reply);
snprintf(buf,64,"%d",j);
reply = redisCommand(c,"LPUSH mylist element-%s", buf);
         freeReplyObject(reply);
    }
/* Let's check what we have inside the list */
reply = redisCommand(c,"LRANGE mylist 0 -1");
if (reply->type == REDIS_REPLY_ARRAY) {
        for (j = 0; j < reply->elements; j++) {
printf("%u) %s\n", j, reply->element[j]->str);
}
}
freeReplyObject(reply);
/* Disconnects and frees the context */
redisFree(c);
return 0;
}
```

З.

```
gcc -o example -g example.c -I /usr/local/include/hiredis -lhiredis
```

4.

example xxx.kvstore.aliyuncs.com 6379 instance\_id password

C/C++ Redis

```
.net客户端
```

```
1.
           .net
    git clone https://github.com/ServiceStack/ServiceStack.Redis
2.
    .net
                   .net
                                  ServiceStack.Redis/lib/tests
3.
4.
         .net
                                           Redis
                                                                     https://github.
  com/ServiceStack/ServiceStack.Redis
   using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Text;
    using System. Threading. Tasks;
    using ServiceStack.Redis;
    namespace ServiceStack.Redis.Tests
    {
             class Program
```

```
public static void RedisClientTest()
```

{

```
string host = "127.0.0.1";/* host */
         string password = "password";/*
                                             */
         RedisClient redisClient = new RedisClient(host, 6379,
password);
         string key = "test-aliyun";
         string value = "test-aliyun-value";
         redisClient.Set(key, value);
         string listKey = "test-aliyun-list";
         System.Console.WriteLine("set key " + key + " value " +
value);
         string getValue = System.Text.Encoding.Default.GetString(
redisClient.Get(key));
         System.Console.WriteLine("get key " + getValue);
         System.Console.Read();
 }
 public static void RedisPoolClientTest()
         string[] testReadWriteHosts = new[] {
         "redis://password@127.0.0.1:6379"/*redis://
                                                          a
                                                                   :
 */
 };
 RedisConfig.VerifyMasterConnections = false;//
 PooledRedisClientManager redisPoolManager = new PooledRedi
sClientManager(10/*
                             */, 10/*
                                                  */, testReadWr
iteHosts);
for (int i = 0; i < 100; i++){</pre>
         IRedisClient redisClient = redisPoolManager.GetClient
();//
         RedisNativeClient redisNativeClient = (RedisNativeClient)
redisClient;
         redisNativeClient.Client = null;//ApsaraDB for Redis
client setname
                                client
                                              null
 try
 {
         string key = "test-aliyun1111";
         string value = "test-aliyun-value1111";
         redisClient.Set(key, value);
         string listKey = "test-aliyun-list";
         redisClient.AddItemToList(listKey, value);
System.Console.WriteLine("set key " + key + " value " +
value);
         string getValue = redisClient.GetValue(key);
         System.Console.WriteLine("get key " + getValue);
         redisClient.Dispose();//
 }catch (Exception e)
 Ł
         System.Console.WriteLine(e.Message);
 }
         System.Console.Read();
 }
 static void Main(string[] args)
 {
         //
         RedisClientTest();
         //
         RedisPoolClientTest();
```

}

### node-redis客户端

1. node-redis

npm install hiredis redis

2. node-redis

Redis

```
var redis = require("redis"),
client = redis.createClient(<port>, <"host">, {detect_buffers: true
});
client.auth("password", redis.print)
```

```
:
```

port

6379 host

```
client = redis.createClient(6379, "r-abcdefg.redis.rds.aliyuncs.com
", {detect_buffers: true});
```

З. Redis

```
//
client.set("key", "OK");
                       String
//
client.get("key", function (err, reply) {
  console.log(reply.toString()); // print `OK`
});
                     Buffer
                                                 Buffer
//
client.get(new Buffer("key"), function (err, reply) {
  console.log(reply.toString()); // print `<Buffer 4f 4b>`
});
client.quit();
```

### C#客户端StackExchange.Redis

- 1. StackExchange.Redis
- 2.

using StackExchange.Redis;

З. ConnectionMultiplexer

ConnectionMultiplexer StackExchange.Redis

// redis config

```
private static ConfigurationOptions configurationOptions =
   ConfigurationOptions.Parse("127.0.0.1:6379, password=xxx, connectTim
   eout=2000");
     //the lock for singleton
    private static readonly object Locker = new object();
     //singleton
    private static ConnectionMultiplexer redisConn;
    //singleton
    public static ConnectionMultiplexer getRedisConn()
         if (redisConn == null)
         {
             lock (Locker)
             {
                 if (redisConn == null || !redisConn.IsConnected)
                 ł
                      redisConn = ConnectionMultiplexer.Connect(
   configurationOptions);
             }
         }
         return redisConn;
    }
            :
   ConfigurationOptions
                                       keepAlive connectRetry name
      StackExchange.Redis.ConfigurationOptions
4. GetDatabase()
                                                ConnectionMultiplexer
    redisConn = getRedisConn();
    var db = redisConn.GetDatabase();
         5
                     demo
                                 API
                                                         String Hash List
  Set SortedSet
                       5

    string

      //set get
     string strKey = "hello";
     string strValue = "world";
     bool setResult = db.StringSet(strKey, strValue);
Console.WriteLine("set " + strKey + " " + strValue + ", result is
       " + setResult);
      //incr
      string counterKey = "counter";
     long counterValue = db.StringIncrement(counterKey);
     Console.WriteLine("incr " + counterKey + ", result is " +
     counterValue);
      //expire
     db.KeyExpire(strKey, new TimeSpan(0, 0, 5));
     Thread.Sleep(5 \times 1000);
     Console.WriteLine("expire " + strKey + ", after 5 seconds, value
     is " + db.StringGet(strKey));
     //mset mget
```

5.

```
Key});
Console.WriteLine("mget " + kv1.Key.ToString() + " " + kv2.Key.
ToString() + ", result is " + values[0] + "&&" + values[1]);
```

hash

```
string hashKey = "myhash";
//hset
db.HashSet(hashKey,"f1","v1");
db.HashSet(hashKey,"f2", "v2");
HashEntry[] values = db.HashGetAll(hashKey);
//hgetall
Console.Write("hgetall " + hashKey + ", result is");
for (int i = 0; i < values.Length;i++)
{
    HashEntry hashEntry = values[i];
    Console.Write(" " + hashEntry.Name.ToString() + " " + hashEntry.
Value.ToString());
}
Console.WriteLine();
```

• list

```
//list key
string listKey = "myList";
//rpush
db.ListRightPush(listKey, "a");
db.ListRightPush(listKey, "b");
db.ListRightPush(listKey, "c");
//lrange
RedisValue[] values = db.ListRange(listKey, 0, -1);
Console.Write("lrange " + listKey + " 0 -1, result is ");
for (int i = 0; i < values.Length; i++)
{
    Console.Write(values[i] + " ");
}
Console.WriteLine();</pre>
```

set

```
//set key
string setKey = "mySet";
//sadd
db.SetAdd(setKey, "a");
db.SetAdd(setKey, "b");
db.SetAdd(setKey, "c");
//sismember
bool isContains = db.SetContains(setKey, "a");
Console.WriteLine("set " + setKey + " contains a is " + isContains
);
```

sortedset

string sortedSetKey = "myZset";
//sadd

```
db.SortedSetAdd(sortedSetKey, "xiaoming", 85);
db.SortedSetAdd(sortedSetKey, "xiaohong", 100);
db.SortedSetAdd(sortedSetKey, "xiaofei", 62);
db.SortedSetAdd(sortedSetKey, "xiaotang", 73);
//zrevrangebyscore
RedisValue[] names = db.SortedSetRangeByRank(sortedSetKey, 0, 2,
Order.Ascending);
Console.Write("zrevrangebyscore " + sortedSetKey + " 0 2, result
is ");
for (int i = 0; i < names.Length; i++)</pre>
   Console.Write(names[i] + " ");
Console.WriteLine();
```

# 7.3 redis-cli连接

{



# 7.4 公网连接

### 前提条件

	PC	Redis			ECS
• Redis		VPC	ECS	Redis	VPC

- Redis
   ECS
   Redis
- ECS Redis
   IP
   ECS PC Redis

ECS

Ê	:			
		Redis		

### ECS Windows篇

Redis ECS

Redis ECS Windows netsh

1. ECS Windows cmd

netsh interface portproxy add v4tov4 listenaddress=ECS IP listenport=6379 connectaddress= Redis connectport= 6379

回管理员:命令提示符 この管理员:命令提示符 このでは、いまである。 こののでは、いまでは、、のからのでは、、のからのでは、、のからのでは、、のからのでは、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、、								
C: VU:	C:\Users\Administrator\netsh interface portproxy show all							
侦听	ipv4:		连接到 ip	pu4:				
地址		端口	地址	2番口				
		6379						
C:\Users\Administrator>netsh interface portproxy delete v4tov4 listenaddress==== listenport=6379								

netsh interface portproxy show all

**ECS** Windows

2.



a. redis-cli ECS Windows IP 1.1.1.1 telnet 1.1.1.1 6379

	b.	ECS windows		Redis	s aı	uth Redis		
	C.							
			PC		ECS	Windows 637	9	
	Redis							
	()							
	port	oroxy						netsh
	port	proxy						
	ро	r tm ap						
З.				netsh	interface	e portproxy	delete	2

v4tov4 listenaddress=ECS IP listenport=6379

### ECS Linux 篇

		Redis	ECS
Redis		ECS Linux	rinetd
1.		ECS Linux	rinetd
	wget   rinetd )	nttp://www.bo .tar.gz&&cd r	utell.com/rinetd/http/rinetd.tar.gz&&tar -xvf inetd sed -i 's/65536/65535/g' rinetd.c ( mkdir /usr/man&&make&&make install
	:		
	meta		
2.		rinetd.conf	
	vi /e <sup>.</sup>	tc/rinetd.con	f
3.			
	0.0.0	.0 6379 Redis	6379 logfile /var/log/rinetd.log
	Ê	:	
		cat /etc/ri	netd.conf
	[root@localh 0.0.0.0 6379 logfile /var	ost rinetd]# cat /etc ) tb.m.c ;/log/rinetd.log	/rinetd.conf nhza.kvstore.aliyuncs.com 6379

/ 7

