## 1.1Units单位

# Redis configuration file example.

# Note on units:when memory size is needed,it is possible to specify

# it in the usual form of 1k 5GB 4M and so forth:

#

# 1k => 1000 bytes

# 1kb => 1024 bytes

# 1m => 1000000 bytes

# 1mb => 1024\*1024 bytes

# 1g => 1000000000 bytes

# 1gb => 1024\*1024\*1024 bytes

#

# units are case insensitive so 1GB 1Gb 1gB are all the same.

1. 配置大小单位，开头定义了一些基本的度量单位，只支持bytes，不支持bit
2. 对大小写不敏感

## 1.2INCLUDES包含

#######################INCLUDES###############################

# Include one or more other config files here.This is useful if you

# have a standard template that goes to all Redis servers but also need

# to customize a few per-server settings.Include files can include

# other files,so use this wisely.

#

# Notice option "include" won't be rewritten by command "CONFIG REWRITE"

# from admin or Redis Sentinel.Since Redis always uses the last processed

# line as value of a configuration directive,you'd better put includes

# at the beginning of this file to avoid overwriting config change at runtime.

#

# If instead you are interested in using includes to override configuration

# options,it is better to use include as the last line.

#

# include /path/to/local.conf

# include /path/to/other.conf

1. 和我们的Struts配置文件类似，可以通过includes包含，redis.conf可以作为总闸，包含其他

## 1.3GENARAL通用

##########################GENERAL#############################

# By default Redis does not run as a daemon.Use 'yes' if you need it.

# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.

daemonize yes 进程后台启动

# If you run Redis from upstart or systemd,Redis can interact with your

# supervision tree. Options:

# supervised no - no supervision interaction

# supervised upstart - signal upstart by putting Redis into SIGSTOP mode

# supervised systemd - signal systemd by writing READY=1 to $NOTIFY\_SOCKET

# supervised auto - detect upstart or systemd method based on

# UPSTART\_JOB or NOTIFY\_SOCKET environment variables

# Note: these supervision methods only signal "process is ready."

# They do not enable continuous liveness pings back to your supervisor.

supervised no

# If a pid file is specified, Redis writes it where specified at startup

# and removes it at exit.

#

# When the server runs non daemonized, no pid file is created if none is

# specified in the configuration. When the server is daemonized, the pid file

# is used even if not specified, defaulting to "/var/run/redis.pid".

#

# Creating a pid file is best effort: if Redis is not able to create it

# nothing bad happens, the server will start and run normally.

pidfile /var/run/redis\_6379.pid

# Specify the server verbosity level.

# This can be one of:

# debug (a lot of information, useful for development/testing)

# verbose (many rarely useful info, but not a mess like the debug level)

# notice (moderately verbose, what you want in production probably)

# warning (only very important / critical messages are logged)

loglevel notice 日志级别

# Specify the log file name. Also the empty string can be used to force

# Redis to log on the standard output. Note that if you use standard

# output for logging but daemonize, logs will be sent to /dev/null

logfile "" 日志名字

# To enable logging to the system logger, just set 'syslog-enabled' to yes,

# and optionally update the other syslog parameters to suit your needs.

# syslog-enabled no 是否把日志输出到syslog中

# Specify the syslog identity.

# syslog-ident redis 指定syslog里的日志标识

# Specify the syslog facility. Must be USER or between LOCAL0-LOCAL7.

# syslog-facility local0 指定syslog设备，值可以是USER或LOCAL0-LOCAL7

# Set the number of databases. The default database is DB 0, you can select

# a different one on a per-connection basis using SELECT <dbid> where

# dbid is a number between 0 and 'databases'-1

databases 16 16个库

# By default Redis shows an ASCII art logo only when started to log to the

# standard output and if the standard output is a TTY. Basically this means

# that normally a logo is displayed only in interactive sessions.

#

# However it is possible to force the pre-4.0 behavior and always show a

# ASCII art logo in startup logs by setting the following option to yes.

always-show-logo yes

#################################NETWORK###################################

# By default, if no "bind" configuration directive is specified, Redis listens

# for connections from all the network interfaces available on the server.

# It is possible to listen to just one or multiple selected interfaces using

# the "bind" configuration directive, followed by one or more IP addresses.

#

# Examples:

#

# bind 192.168.1.100 10.0.0.1

# bind 127.0.0.1 ::1

#

# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the

# internet, binding to all the interfaces is dangerous and will expose the

# instance to everybody on the internet. So by default we uncomment the

# following bind directive, that will force Redis to listen only into

# the IPv4 loopback interface address (this means Redis will be able to

# accept connections only from clients running into the same computer it

# is running).

#

# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES

# JUST COMMENT THE FOLLOWING LINE.

# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

bind 127.0.0.1

# Protected mode is a layer of security protection, in order to avoid that

# Redis instances left open on the internet are accessed and exploited.

#

# When protected mode is on and if:

#

# 1) The server is not binding explicitly to a set of addresses using the

# "bind" directive.

# 2) No password is configured.

#

# The server only accepts connections from clients connecting from the

# IPv4 and IPv6 loopback addresses 127.0.0.1 and ::1, and from Unix domain

# sockets.

#

# By default protected mode is enabled. You should disable it only if

# you are sure you want clients from other hosts to connect to Redis

# even if no authentication is configured, nor a specific set of interfaces

# are explicitly listed using the "bind" directive.

protected-mode yes

# Accept connections on the specified port, default is 6379 (IANA #815344).

# If port 0 is specified Redis will not listen on a TCP socket.

port 6379 端口号

# TCP listen() backlog.

#

# In high requests-per-second environments you need an high backlog in order

# to avoid slow clients connections issues. Note that the Linux kernel

# will silently truncate it to the value of /proc/sys/net/core/somaxconn so

# make sure to raise both the value of somaxconn and tcp\_max\_syn\_backlog

# in order to get the desired effect.

tcp-backlog 511 连接队列

# Unix socket.

#

# Specify the path for the Unix socket that will be used to listen for

# incoming connections. There is no default, so Redis will not listen

# on a unix socket when not specified.

#

# unixsocket /tmp/redis.sock

# unixsocketperm 700

# Close the connection after a client is idle for N seconds (0 to disable)

timeout 0 客户端空闲N秒后关闭连接（0表示禁用）

# TCP keepalive.

#

# If non-zero, use SO\_KEEPALIVE to send TCP ACKs to clients in absence

# of communication. This is useful for two reasons:

#

# 1) Detect dead peers.

# 2) Take the connection alive from the point of view of network

# equipment in the middle.

#

# On Linux, the specified value (in seconds) is the period used to send ACKs.

# Note that to close the connection the double of the time is needed.

# On other kernels the period depends on the kernel configuration.

#

# A reasonable value for this option is 300 seconds, which is the new

# Redis default starting with Redis 3.2.1.

tcp-keepalive 300

tcp-backlog:

设置tcp的backlog，backlog其实是一个连接队列，backlog队列总和=未完成三次握手队列+已经完成三次握手队列。

在高并发环境下你需要一个高backlog值来避免慢客户端连接问题。注意Linux内核会将这个值减小到/proc/sys/net/core/somaxconn的值，所以需要确认增大somaxconn和tcp\_max\_syn\_backlog两个值来达到想要的效果

tcp-keepalive：

单位为秒，如果设置为0，则不会进行keepalive检测，建议设置成60。

## 1.4SNAPSHOTTING快照

################################SNAPSHOTTING##############################

#

# Save the DB on disk:

#

# save <seconds> <changes>

#

# Will save the DB if both the given number of seconds and the given

# number of write operations against the DB occurred.

#

# In the example below the behaviour will be to save:

# after 900 sec (15 min) if at least 1 key changed

# after 300 sec (5 min) if at least 10 keys changed

# after 60 sec if at least 10000 keys changed

#

# Note: you can disable saving completely by commenting out all "save" lines.

#

# It is also possible to remove all the previously configured save

# points by adding a save directive with a single empty string argument

# like in the following example:

#

# save ""

save 900 1

save 300 10

save 60 10000

# By default Redis will stop accepting writes if RDB snapshots are enabled

# (at least one save point) and the latest background save failed.

# This will make the user aware (in a hard way) that data is not persisting

# on disk properly, otherwise chances are that no one will notice and some

# disaster will happen.

#

# If the background saving process will start working again Redis will

# automatically allow writes again.

#

# However if you have setup your proper monitoring of the Redis server

# and persistence, you may want to disable this feature so that Redis will

# continue to work as usual even if there are problems with disk,

# permissions, and so forth.

stop-writes-on-bgsave-error yes

# Compress string objects using LZF when dump .rdb databases?

# For default that's set to 'yes' as it's almost always a win.

# If you want to save some CPU in the saving child set it to 'no' but

# the dataset will likely be bigger if you have compressible values or keys.

rdbcompression yes

# Since version 5 of RDB a CRC64 checksum is placed at the end of the file.

# This makes the format more resistant to corruption but there is a performance

# hit to pay (around 10%) when saving and loading RDB files, so you can disable it

# for maximum performances.

#

# RDB files created with checksum disabled have a checksum of zero that will

# tell the loading code to skip the check.

rdbchecksum yes

# The filename where to dump the DB

dbfilename dump.rdb

# The working directory.

#

# The DB will be written inside this directory, with the filename specified

# above using the 'dbfilename' configuration directive.

#

# The Append Only File will also be created inside this directory.

#

# Note that you must specify a directory here, not a file name.

dir ./

save：

秒钟 写操作次数

RDB是整个内存的压缩过的Snapshot，RDB的数据结构，可以配置复合的快照触发条件，默认：

是1分钟内改了1万次，

或5分钟内改了10次，

或15分钟内改了1次。

如果想禁用RDB持久化的策略，只要不设置任何save指令，或者给save传入一个空字符串参数也可以。

stop-writes-on-bgsave-error:

如果配置成no，表示你不在乎数据不一致或者有其他的手段发现和控制。

redbcompression：

对于存储到磁盘中的快照，可以设置是否进行压缩存储。如果是的话，redis会采用LZF算法进行压缩。如果你不想消耗CPU来进行压缩的话，可以设置为关闭此功能。

rdbchecksum：

在存储快照后，还可以让redis使用CRC64算法来进行数据校验，但是这样做会增加大约10%的性能消耗，如果希望获取到最大的性能提升，可以关闭此功能。

## 1.5REPLICATION复制

## 1.6SECURITY安全

访问密码的查看、设置和取消。

#################################SECURITY##################################

# Require clients to issue AUTH <PASSWORD> before processing any other

# commands. This might be useful in environments in which you do not trust

# others with access to the host running redis-server.

#

# This should stay commented out for backward compatibility and because most

# people do not need auth (e.g. they run their own servers).

#

# Warning: since Redis is pretty fast an outside user can try up to

# 150k passwords per second against a good box. This means that you should

# use a very strong password otherwise it will be very easy to break.

#

# requirepass foobared

# Command renaming.

#

# It is possible to change the name of dangerous commands in a shared

# environment. For instance the CONFIG command may be renamed into something

# hard to guess so that it will still be available for internal-use tools

# but not available for general clients.

#

# Example:

#

# rename-command CONFIG b840fc02d524045429941cc15f59e41cb7be6c52

#

# It is also possible to completely kill a command by renaming it into

# an empty string:

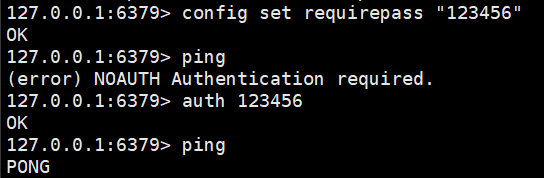
#

# rename-command CONFIG ""

#

# Please note that changing the name of commands that are logged into the

# AOF file or transmitted to replicas may cause problems.



## 1.7LIMITS限制

#################################CLIENTS###################################

529

530 # Set the max number of connected clients at the same time.By default

531 # this limit is set to 10000 clients,however if the Redis server is not

532 # able to configure the process file limit to allow for the specified limit

533 # the max number of allowed clients is set to the current file limit

534 # minus 32 (as Redis reserves a few file descriptors for internal uses).

535 #

536 # Once the limit is reached Redis will close all the new connections sending

537 # an error 'max number of clients reached'.

538 #

539 # maxclients 10000

540

541##########################MEMORY MANAGEMENT#############################

542

543 # Set a memory usage limit to the specified amount of bytes.

544 # When the memory limit is reached Redis will try to remove keys

545 # according to the eviction policy selected (see maxmemory-policy).

546 #

547 # If Redis can't remove keys according to the policy, or if the policy is

548 # set to 'noeviction', Redis will start to reply with errors to commands

549 # that would use more memory, like SET, LPUSH, and so on, and will continue

550 # to reply to read-only commands like GET.

551 #

552 # This option is usually useful when using Redis as an LRU or LFU cache, or t o

553 # set a hard memory limit for an instance (using the 'noeviction' policy).

554 #

555 # WARNING: If you have replicas attached to an instance with maxmemory on,

# the size of the output buffers needed to feed the replicas are subtracted

557 # from the used memory count, so that network problems / resyncs will

558 # not trigger a loop where keys are evicted, and in turn the output

559 # buffer of replicas is full with DELs of keys evicted triggering the deletio n

560 # of more keys, and so forth until the database is completely emptied.

561 #

562 # In short... if you have replicas attached it is suggested that you set a lo wer

563 # limit for maxmemory so that there is some free RAM on the system for replic a

564 # output buffers (but this is not needed if the policy is 'noeviction').

565 #

566 # maxmemory <bytes>

567

568 # MAXMEMORY POLICY: how Redis will select what to remove when maxmemory

569 # is reached. You can select among five behaviors:

570 #

571 # volatile-lru -> Evict using approximated LRU among the keys with an expire set.

572 # allkeys-lru -> Evict any key using approximated LRU.

573 # volatile-lfu -> Evict using approximated LFU among the keys with an expire set.

574 # allkeys-lfu -> Evict any key using approximated LFU.

575 # volatile-random -> Remove a random key among the ones with an expire set.

576 # allkeys-random -> Remove a random key, any key.

577 # volatile-ttl -> Remove the key with the nearest expire time (minor TTL)

578 # noeviction -> Don't evict anything, just return an error on write operations.

579 #

580 # LRU means Least Recently Used

581 # LFU means Least Frequently Used

582 #

583 # Both LRU, LFU and volatile-ttl are implemented using approximated

584 # randomized algorithms.

585 #

586 # Note: with any of the above policies, Redis will return an error on write

587 # operations, when there are no suitable keys for eviction.

588 #

589 # At the date of writing these commands are: set setnx setex append

590 # incr decr rpush lpush rpushx lpushx linsert lset rpoplpush sadd

591 # sinter sinterstore sunion sunionstore sdiff sdiffstore zadd zincrby

592 # zunionstore zinterstore hset hsetnx hmset hincrby incrby decrby

593 # getset mset msetnx exec sort

594 #

595 # The default is:

596 #

597 # maxmemory-policy noeviction 默认永不过期

598

599 # LRU,LFU and minimal TTL algorithms are not precise algorithms but approximated

600 # algorithms (in order to save memory), so you can tune it for speed or

601 # accuracy. For default Redis will check five keys and pick the one that was

602 # used less recently, you can change the sample size using the following

603 # configuration directive.

604 #

605 # The default of 5 produces good enough results. 10 Approximates very closely

606 # true LRU but costs more CPU. 3 is faster but not very accurate.

607 #

608 # maxmemory-samples 5 设置样本数量

609

610 # Starting from Redis 5, by default a replica will ignore its maxmemory setti ng

611 # (unless it is promoted to master after a failover or manually). It means

612 # that the eviction of keys will be just handled by the master, sending the

613 # DEL commands to the replica as keys evict in the master side.

614 #

615 # This behavior ensures that masters and replicas stay consistent, and is usu ally

616 # what you want, however if your replica is writable, or you want the replica to have

617 # a different memory setting, and you are sure all the writes performed to th e

618 # replica are idempotent, then you may change this default (but be sure to un derstand

619 # what you are doing).

620 #

621 # Note that since the replica by default does not evict, it may end using mor e

622 # memory than the one set via maxmemory (there are certain buffers that may

623 # be larger on the replica, or data structures may sometimes take more memory and so

624 # forth). So make sure you monitor your replicas and make sure they have enou gh

625 # memory to never hit a real out-of-memory condition before the master hits

626 # the configured maxmemory setting.

627 #

628 # replica-ignore-maxmemory yes

内存淘汰策略：



## 1.8APPEND ONLY MODE追加

## 1.9常见配置redis.conf介绍

