

前言

如何给Redis设置密码，以防止其它未经授权的客户端进行连接呢？
怎么知道哪些命令执行的比较慢呢？

本文将带你熟悉Redis管理方面的知识，包含安全和通信协议等等内容。
与此同时，还会介与之紧密相关的第三方管理工具。

上一篇博客《[Redis真的又小又快又持久吗](#)》，其实只能作入门指南来看，并没有多少深度，但是对于面试有不少帮助的。标题有噱头才会引起更多爱好者多Redis的探索，进一步走进Redis这个五彩斑斓的世界，进阶知识只有深入学习才能更快掌握。



正文

本文将带你熟悉Redis管理方面的知识，包含安全和通信协议等等内容。
与此同时，还会介与之紧密相关的第三方管理工具。

一、安全

谈到安全，我们会联想到些什么？

比如，可信任的环境会给我们带来安全感，陌生的环境则会让你感到未知的恐惧和孤独。

再比如，国产化替代信创项目（安可替代），这里我简称为国创项目，就是要达到信任、安全可靠以及自主可控的目的。

上面谈了这么多（瞎扯了很多，我黔驴尽穷了），只是为了提升我们的安全意识。

Redis以简洁为美，创始人曾这么描述过。但同样在安全层面也没做过多的工作。

这里补充一点，上次没有讲到如何**优雅的关闭Redis服务**。虽然可以杀掉进程来控制，但推荐使用如下方式关闭：

```
$ /opt/redis-6.0.8/src/redis-cli shutdown
```

1、可信环境

Redis的安全设计是基于“Redis运行在可信任的环境”这个前提下做出来的。在生产环境（正式发布环境）运行时，不允许外部直接连接到Redis服务器上，此时应该通过应用程序进行中转，运行在可信任的环境中是保证Redis安全至关重要的方法。

1.1、bind参数

在Redis的默认配置文件redis.conf中，只会接受本地的网络请求。但通过在配置文件中修改 bind 参数更改这一设置，默认的bind设置为：

```
# ~~~ 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 默认绑定的IP地址
```

```
bind:127.0.0.1
```

bind 参数同样可以绑定多个IP地址，IP地址以间隔空格分隔，如下示例：

```
# 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
```

```
# Examples:
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
```

1.2、protected-mode参数

在Redis3.2的版本中，引入了一个特殊模式：保护模式，来更好地确保Redis运行在可信环境之中。值得注意的是，保护模式在默认情况下是开启的。

参数设置：

```
#开启保护模式
protected-mode yes
#禁止保护模式
protected-mode no
```

```
#
# 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
```

默认开启保护模式，设置为no则禁止

作用：

- 开启保护模式：接收到来自不在bind绑定的网络客户端发送命令时，如果客户端没有设置密码，Redis会返回错误拒绝（DENIED）执行该命令。
- 禁止保护模式：可以在配置中使用 `protected-mode no` 禁止。
- 安全：对于生产环境需要确保开启了护盾（防火墙），达到确保可信客户端连接服务器的目的。

在测试的时候，比如我在Windows下连接我的linux上的Redis服务。为了方便测试，此时临时关闭防护墙firewalld，或者采用 `firewall-cmd` 命令加入6379默认端口以及Redis服务，关于防火墙的知识可以参考我之前的文章《[firewalld与iptables防火墙工具](#)》：

```
#临时关闭防火墙
systemctl stop firewalld.service
```

注意：Redis3.2之前的版本默认会绑定所有网络接口，任何网络上的计算机（包含公网）都可连接至Redis服务器上。使用旧版的需要注意，最好修改这个参数，或者升级到新版。

2、数据库密码

Redis中提供了数据库密码功能。最开始我傻傻的以为直接就能连上，岂不是没有密码，真不安全。直到后来在工作的实践中，才发现原来这货可以是设置密码的，只是我以前并不知道而已。值得注意的是：并且在6.0版本中支持多用户权限控制功能。

2.1、Redis密码设置

在我的上一篇文章也有提到过。Redis数据库密码是通过参数 `requirepass` 来控制的，默认的6.0.8版本是禁用掉了，需要手动开启。

```
# IMPORTANT NOTE: starting with Redis 6 "requirepass" is just a compatibility
# layer on top of the new ACL system. The option effect will be just setting
# the password for the default user. Clients will still authenticate using
# AUTH <password> as usually, or more explicitly with AUTH default <password>
# if they follow the new protocol: both will work.
#
#requirepass foobared
requirepass 123456
```

默认是禁用掉的，去掉#则开启
然后设置自己的密码。

```
#默认禁用掉了
#requirepass foobared
#启用密码
requirepass 123456
```

客户端每次连接到Redis时都需要发送密码，否则Redis会拒绝执行客户端发来的命令。例如我使用Windows客户端连接：

开启了保护模式，开始提示 `DENIED`。利用 `bind` 绑定了信任的ip或者禁止保护模式，最后还会提示密码为验证。


```
# If the master is password protected (using the "requirepass" configuration
# directive below) it is possible to tell the replica to authenticate before
# starting the replication synchronization process, otherwise the master will
# refuse the replica request.
#
# masterauth <master-password>
```

主库开启密码，
从库需要开启这个配置验证

```
masterauth <master-password>
```

3、命名命令

Redis支持在配置文件中将命令重命名，例如将FLUSHALL命令重命名为一个比较复杂的名字，达到保证只有自己的应用可以使用该命令。当然，这个功能可以看做在6.0版本之前没有ACL，作为对命令安全性的一个补充。如下配置：

```
rename-command FLUSHALL redisabcdsky1r2d3is
```

如果希望直接一点，直接禁用，通过重命名为空字符

```
rename-command FLUSHALL ""
```

再次强调：安全起见，无论设置密码还是重命名命令，都应遵循保证配置文件的安全性，否则就无意义了。

二、通信协议

之前有了解到Redis的主从复制以及持久化AOF文件的格式，通过了解Redis通信协议能更好的理解Redis。

当然Redis支持两种通信协议。如下：

- 一种是二进制安全的统一**请求协议 (unified request protocol)**
- 第二种是比较直观的便于在telnet程序中输入的**简单协议**

1、简单协议

简单协议适合在telnet程序中和Redis通信。如下是通过telnet测试与Redis通信：

linux下Redhat系列安装telnet通过**yum命令**：

```
yum -y install telnet
```

Windows在**启用或关闭Windows功能**中启用telnet

```
[root@dywangk redis-6.0.8]# telnet 127.0.0.1 6379
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^['.
auth 123456 #同样需要验证密码,之前设置了密码
+OK
set foo bar
+OK
get foo
$3
bar
#输入quit退出telnet
```

1.1、错误回复

错误回复 (error reply) 以 - 开头并在后面跟着错误信息：

```
-ERR unknown command ``, with args beginning with:
```

1.2、状态回复

状态回复 (status reply) 以+开头

```
+OK
```

1.3、整数回复

整数回复 (integer reply) 以：开头

```
:3
```

1.4、字符串回复

字符串 (bulk reply) 回复以\$开头

```
$3
```

2、统一请求协议

统一请求协议是从Redis1.2开始加入的，其命令格式与多行字符串回复格式类似。也以telnet为例演示：

```
[root@dywangk redis-6.0.8]# telnet 127.0.0.1 6379
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^['.
auth 123456 #同样需要验证密码,之前设置了密码
+OK
*3
$3
set
$3
foo
$3
```

```
bar
+OK
#输入quit退出telnet
```

同样，在发送命令的时候指定了后面字符串的长度，所以每个命令的每个参数都可以包含二进制的字符。

Redis的AOF文件和主从复制时数据库发送的内容使用了统一请求协议。如果简单的使用telnet与Redis进行通信，使用简单协议即可。

三、管理工具

1、redis-cli

看到redis-cli大家肯定不陌生，是的我们学习测试快速融入都是使用的redis-cli命令进行的，Redis自带的客户端。Redis可以执行大部分的Redis命令，包括查看数据库信息的 `info` 命令、更改数据库设置的 `config` 命令和强制进行RDB快照的 `save` 命令。简单介绍几个管理Redis常用的命令。

1.1、耗时命令日志

当一条命令执行时间超过限制时，Redis会将该命令的执行时间等信息加入耗时命令日志（slow log）以供开发者查看。通过配置文件的 `slowlog-log-slower-than 10000` 参数设置限制，注意单位是微秒，可以看到默认为10000。通过 `slowlog-max-len 128` 限制记录的条数。

获取当前耗时命令日志

```
slowlog get
```

每条日志由以下4个部分组成

- 唯一日志ID
- 执行的Unix时间
- 耗时时间，单位为微秒
- 命令及其参数

测试时，将slowlog-log-slower-than 0 参数设置为0

```
slowlog-log-slower-than 0
```

1.2、命令监控

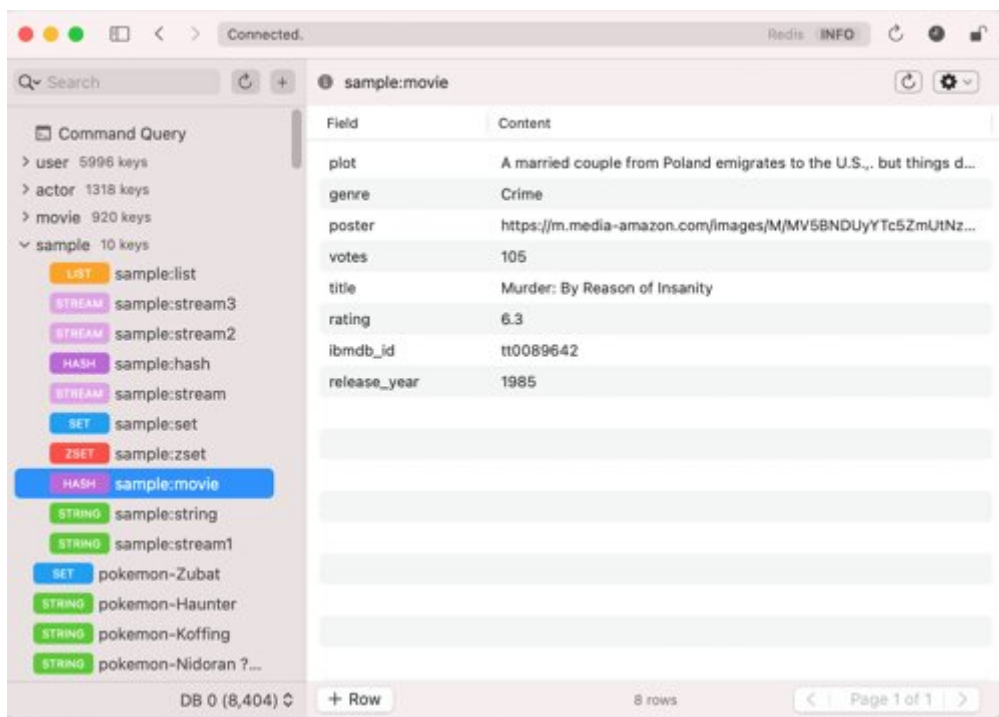
Redis提供了monitor来监控Redis执行的所有命令，redis-cli也支持。例如：

```
monitor
```

注意：一般用于调试和纠错使用。

2、Medis

获取地址：<https://getmedis.com/>



当Redis中的键比较多时，此时使用redis-cli管理略显不足。Redis是一款macOS下的可视化Redis管理工具。通过界面即可实现管理Redis。

3、phpRedisAdmin

看到phpRedisAdmin，大家也许会联想到以网页形式管理MySQL的phpMyAdmin管理工具。

下载地址：<https://github.com/erikdubbelboer/phpRedisAdmin>

关于工具的使用，可以参考github说明，这里不做过多介绍。

建议：github那访问速度大家都懂的，建议导入到gitee作为镜像仓库使用，每隔一段时间同步。

4、Rdbtools

一款采用Python语言开发的Redis的快照文件解析器，它可以根据快照文件导出json数据文件、分析Redis中每个键的占用空间情况。

下载地址：<https://github.com/sripathikrishnan/redis-rdb-tools>

关于工具的使用，可以参考github说明，这里不做过多介绍。

5、命令参考

最后介绍一个Redis命令大全参考网站，**源自于Redis官网**，链接如下：

<https://redis.io/commands>