

Bitmaps（位图）、HyperLogLog、GEO（地理信息定位）等多种数据结构和算法组成，因此 Redis 可以满足很多的应用场景。因为 Redis 会将所有数据都存放再内存中，所以它的读写性能非常惊人。不仅如此，Redis 还可以将内存的数据利用快照和日志的形式保存到硬盘上，这样在发生类似断电或者机器故障的时候，内存中的数据不会“丢失”。除了上述功能以外，Redis 还提供了键过期、发布订阅、事务、流水线、Lua 脚本等附加功能。总之，如果在合适的场景使用 Redis，它就会像一把瑞士军刀一样所向披靡。

2. Redis 的使用场景

2.1 缓存（Cache）

缓存机制几乎在所有大型网站都有使用，合理地使用缓存不仅可以加速数据的访问速度，而且能够有效地降低后端数据源的压力。Redis 提供了键值过期时间设置，并且也提供了灵活控制最大内存和内存溢出后的淘汰策略。可以这么说，一个合理的缓存设计能够为一个网站的稳定保驾护航。

2.2 排行榜系统

排行榜系统几乎存在于所有的网站，例如按照热度排名的排行榜，按照发布时间的排行榜，按照各种复杂维度计算出的排行榜，Redis 提供了列表和有序集合的结构，合理地使用这些数据结构可以很方便地构建各种排行榜系统。

2.3 计数器应用

计数器在网站中的作用至关重要，例如视频网站有播放数、电商网站有浏览数，为了保证数据的实时性，每一次播放和浏览都要做加 1 的操作，如果并发量很大对于传统关系型数据的性能是一种挑战。Redis 天然支持计数功能而且计数的性能也非常好，可以说是计数器系统的重要选择。

2.4 社交网络

赞 / 踩、粉丝、共同好友 / 喜好、推送、下拉刷新等是社交网站的必备功能，由于社交网站访问量通常比较大，而且传统的关系型数据不太合适保存这种类型的数据，Redis 提供的数据结构可以相对比较容易地实现这些功能。

2.5 消息队列系统

消息队列系统可以说是一个大型网站的必备基础组件，因为其具有业务解耦、非实时业务削峰等特性。Redis 提供了发布订阅功能和阻塞队列的功能，虽然和专业的消息队列比还不够足够强大，但是对于一般的消息队列功能基本可以满足。

3. Redis 的基本操作

3.1 Redis 安装

CentOS 安装Redis

- 1 #使用yum安装Redis
- 2 yum -y install redis

Ubuntu 安装Redis

- 1 #使用apt安装Redis
- 2 apt install redis -y

3.2 Redis 启动/停止

CentOS

- 1 #查看是否启动
- 2 ps -ef|grep redis
- 3
- 4 #启动redis
- 5 systemctl start redis
- 6
- 7 #停止Redis
- 8 systemctl stop redis
- 9
- 10 #重启Redis
- 11 systemctl restart redis

Ubuntu

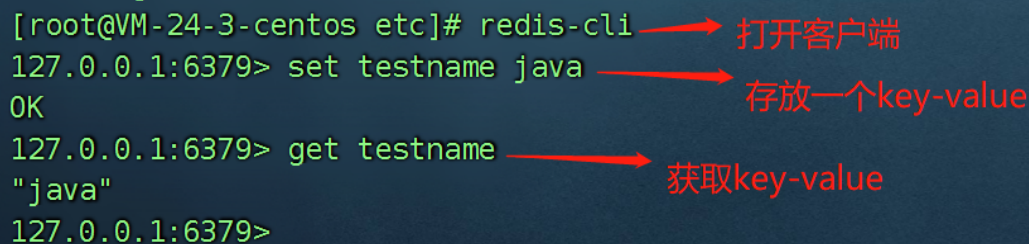
- 1 #查看Redis状态
- 2 systemctl status redis-server
- 3
- 4 #启动redis
- 5 service redis-server start
- 6
- 7 #停止redis服务
- 8 service redis-server stop
- 9
- 10 #重启redis服务
- 11 service redis-server restart

3.3 操作 Redis

使用以下命令启动redis客户端：

```
1 redis-cli
```

具体操作如下图所示：



```
[root@VM-24-3-centos etc]# redis-cli
127.0.0.1:6379> set testname java
OK
127.0.0.1:6379> get testname
"java"
127.0.0.1:6379>
```

Annotations (in red):

- redis-cli → 打开客户端
- set testname java → 存放一个key-value
- get testname → 获取key-value

常见命令介绍：

- SET key value
- GET key
- DEL key [key ...]
- EXISTS key [key ...]
- EXPIRE key seconds
- KEYS pattern

更多命令: <https://redis.io/commands/set/>

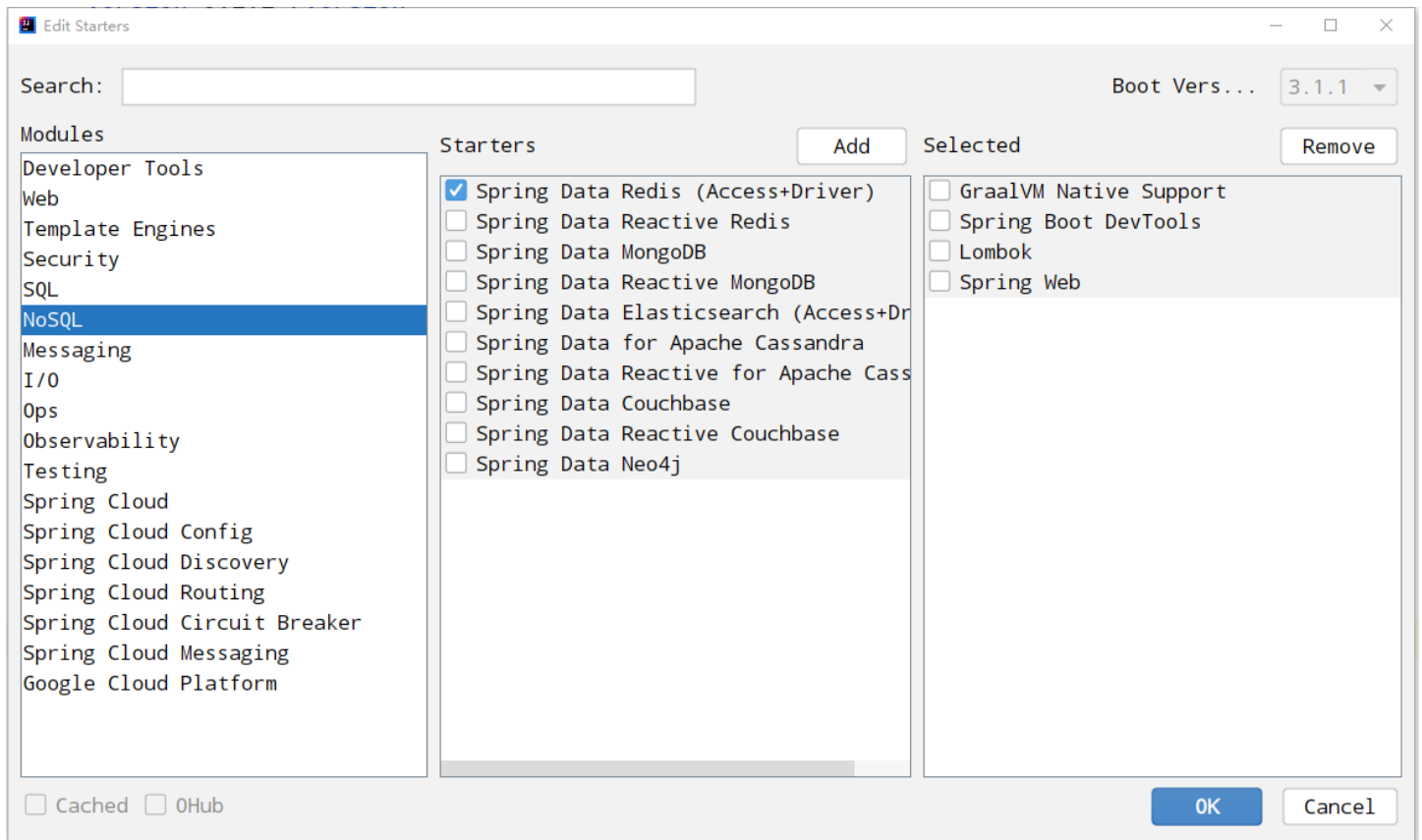
4. Redis使用

4.1 SpringBoot集成Redis

4.1.1 添加依赖

```
1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-data-redis</artifactId>
4 </dependency>
```

或者使用快速面板添加Redis依赖



4.1.2 配置Redis

```
1 spring:
2   data:
3     redis:
4       host: 127.0.0.1
5       port: 6379
6       timeout: 60s #连接空闲超过N(s秒、ms毫秒)后关闭, 0为禁用, 这里配置值和tcp-
keepalive值一致
7     lettuce:
8       pool:
9         max-active: 8 #允许最大连接数
10        max-idle: 8 #最大空闲连接数, 默认8
11        min-idle: 0 #最小空闲连接数
12        max-wait: 5s #请求获取连接等待时间
```

注意!!!

SpringBoot2.X 和SpringBoot 3.X的配置不一样

Spring Boot2.X配置参考:

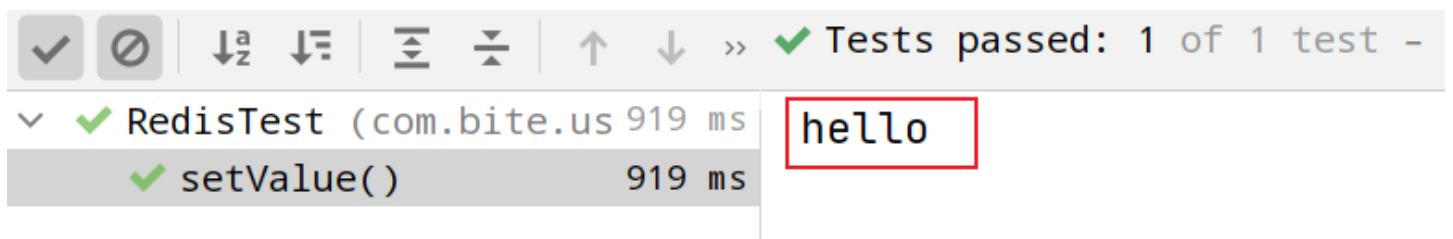
```
1 spring:
2   redis:
3     database: 1
```

```
4     port: 6379
5     host: 127.0.0.1
```

4.1.3 测试代码

```
1 @SpringBootTest
2 public class RedisTest {
3     @Autowired
4     private StringRedisTemplate redisTemplate;
5
6     @Test
7     void setValue(){
8         redisTemplate.opsForValue().set("key", "hello");
9         String value = redisTemplate.opsForValue().get("key");
10        System.out.println(value);
11    }
12 }
```

运行程序(需要配置Redis端口转发, 才能执行成功)



4.2 Redis配置端口转发

Redis 服务器安装在云服务器上, 我们的代码在本地主机.

要想让本地主机能访问 redis, 有两种方式:

1. 开放Redis端口号[不推荐]

把 redis 的端口通过云服务器后台页面的 "防火墙" / "安全组" 放开端口到公网上, 但是这个操作非常危险(黑客会顺着 redis 端口进来).

2. 端口转发[推荐]

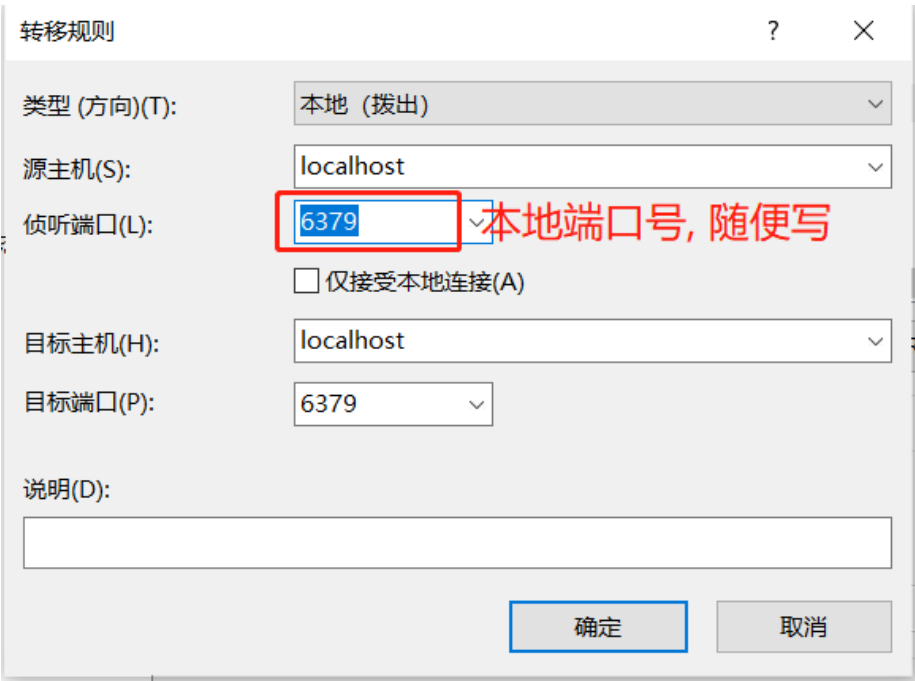
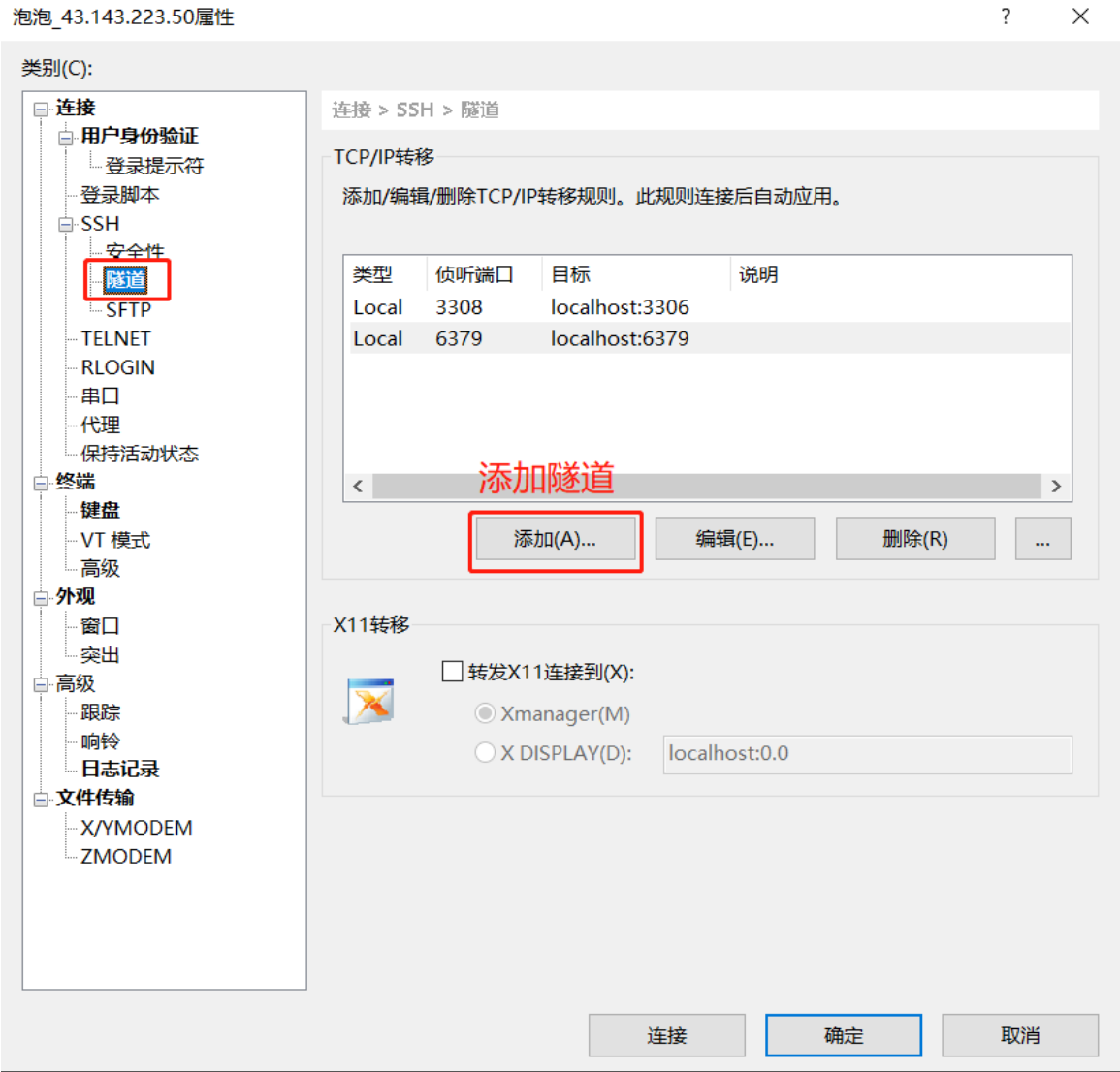
我们可以使用端口转发的方式, 直接把服务器的 redis 端口映射到本地.

具体操作:

在 xshell 中, 进行如下配置:

1) 右键云服务器的会话, 选择属性.

2) 找到隧道 -> 配置转移规则.



3) 使用该会话连接服务器.

全部配置之后, 需要重新关闭连接, 打开连接才可以

后续访问云服务器的Redis, 就可以使用 127.0.0.1 6379端口号(上图中配置的端口号) 来访问

4.3 Redis客户端

除了通过云服务器查看Redis, 市面上也有一些客户端或者插件可以操作Redis

这两种方式分别各介绍一种

1. 客户端 Another-Redis-Desktop-Manager

下载地址: [Another Redis Desktop Manager](#)

安装后创建连接即可

Edit Connection

* Host

127.0.0.1

* Port

6379

Password

Auth

Username

ACL in Redis >= 6.0

Connection Name

127.0.0.1@6379

Separator

:

☐ SSH

☐ SSL

☐ Sentinel ?

☐ Cluster ?

☐ Readonly ?

Cancel

OK

端口号设置为上面配置的监听端口号即可.

New Connection

127.0.0.1@6379

DB0

+ New Key

Enter To Search

key

127.0.0.1@6379

key | 127.0.0.1@6379 | DB0

String

key

Text

Size: 5B

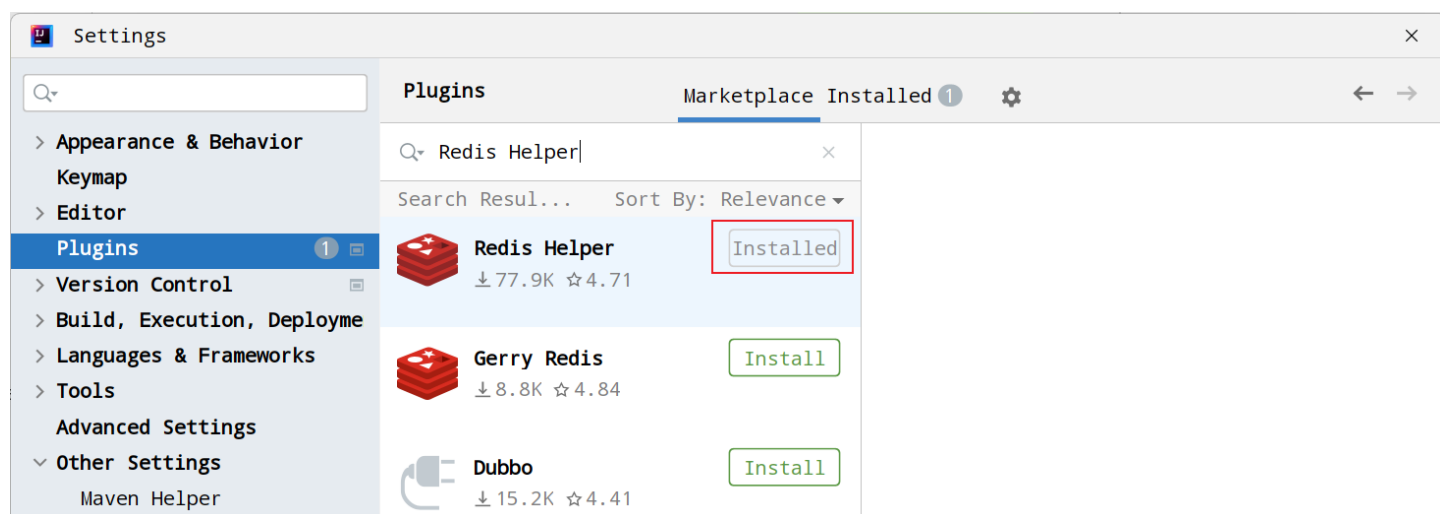
Copy

hello

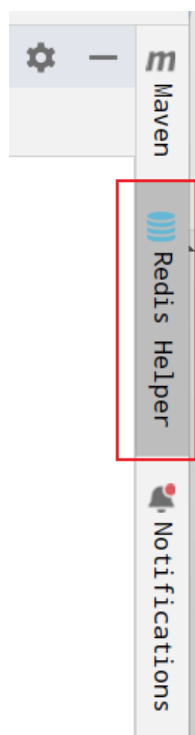
可以查看到刚才存储的数据

2. 插件

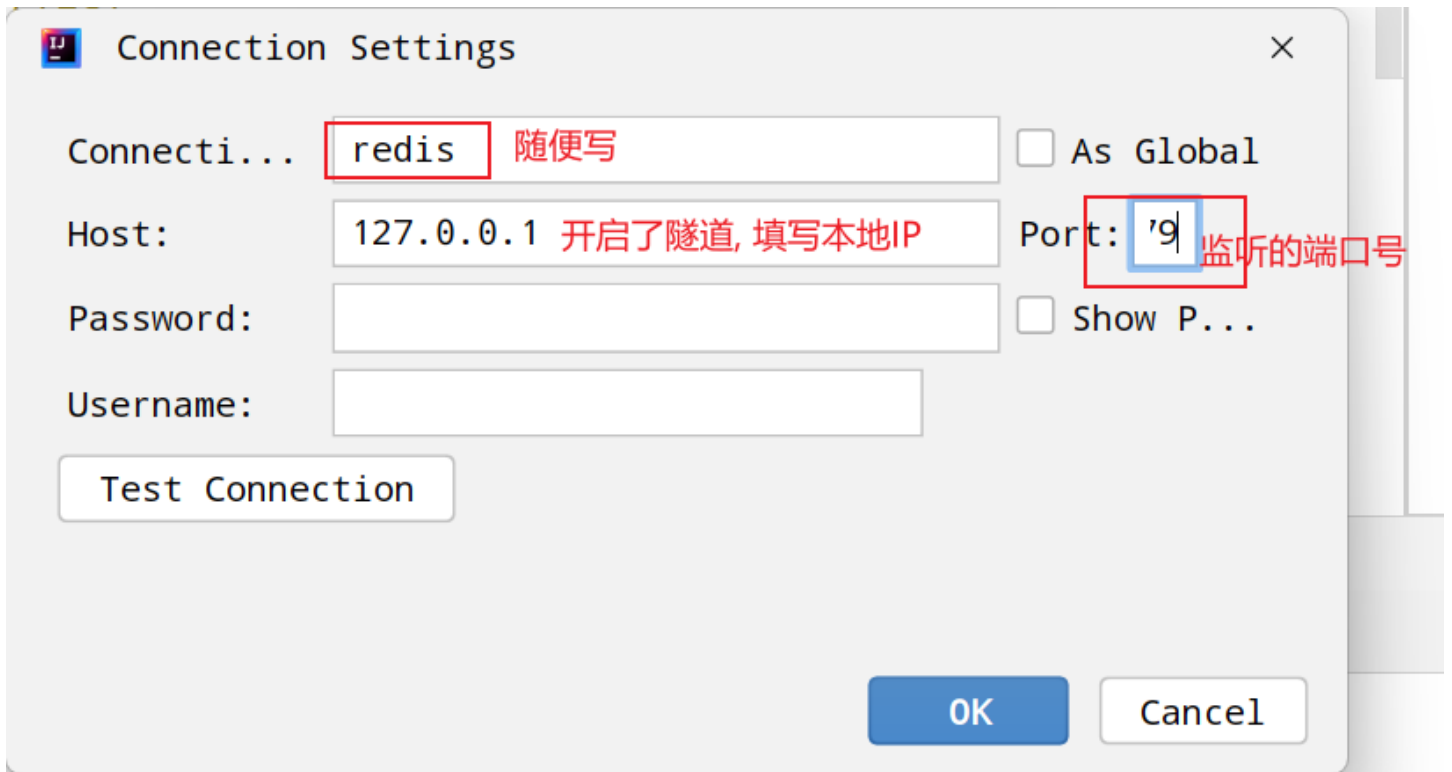
在File->Settings->Plugins->搜索Redis Helper插件



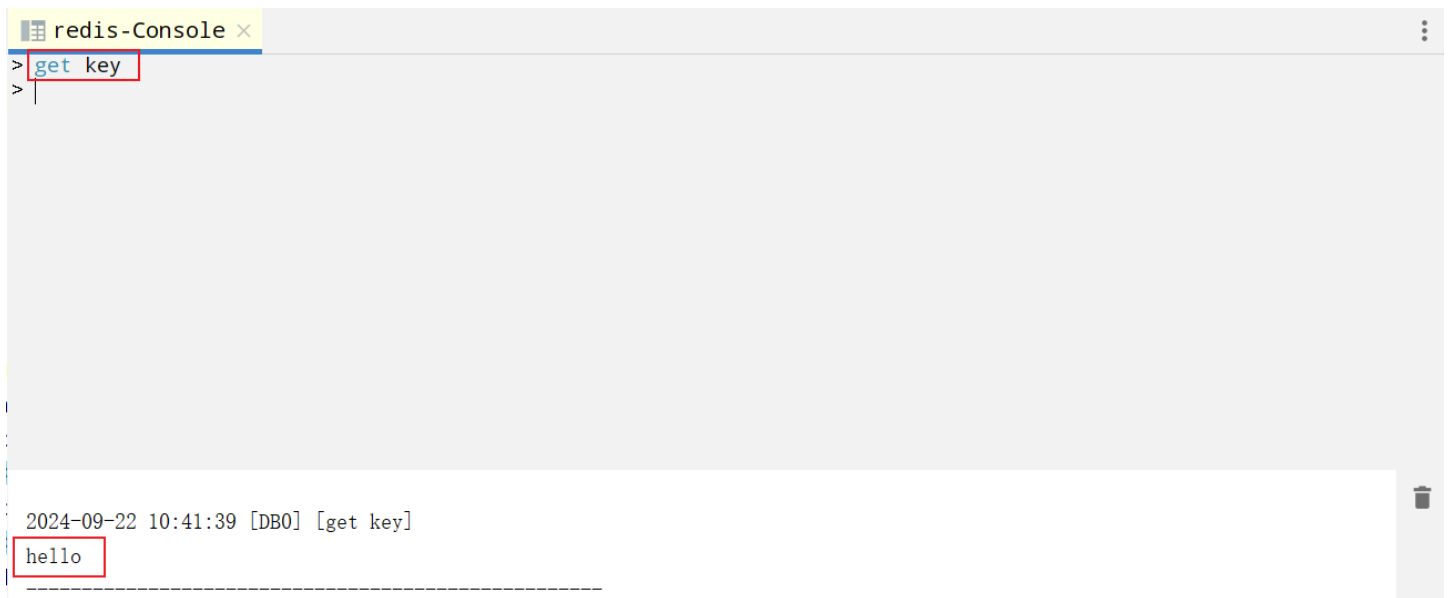
右侧会生成一个页签Redis Helper



然后点击左上角+, 登录远程Redis



操作redis



也提供了界面

Filter: Group by:

DB0 (1)

key

String:

Rename key

Reload value

Value size: 5bytes

View as:

Save

1 hello

redis

- DB0 (1)
- DB1 (0)
- DB2 (0)
- DB3 (0)
- DB4 (0)
- DB5 (0)
- DB6 (0)
- DB7 (0)
- DB8 (0)
- DB9 (0)
- DB10 (0)
- DB11 (0)
- DB12 (0)
- DB13 (0)
- DB14 (0)
- DB15 (0)