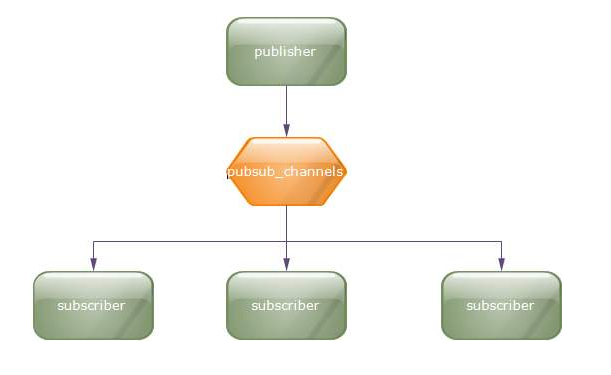
# Redis\_02

# 今日目标4

* 能够了解快照(rdb)持久化与aof持久化生成文件的扩展区别
* 能够给redis设置验证密码
* 能够安装好php的redis扩展
* 能够使用php连接成功redis服务

# 一、发布与订阅

Redis发布订阅(publish/subscribe)是一种消息通信模式，发送者(publish)发送消息，订阅者(subscribe)订阅后接受频道消息。



订阅频道

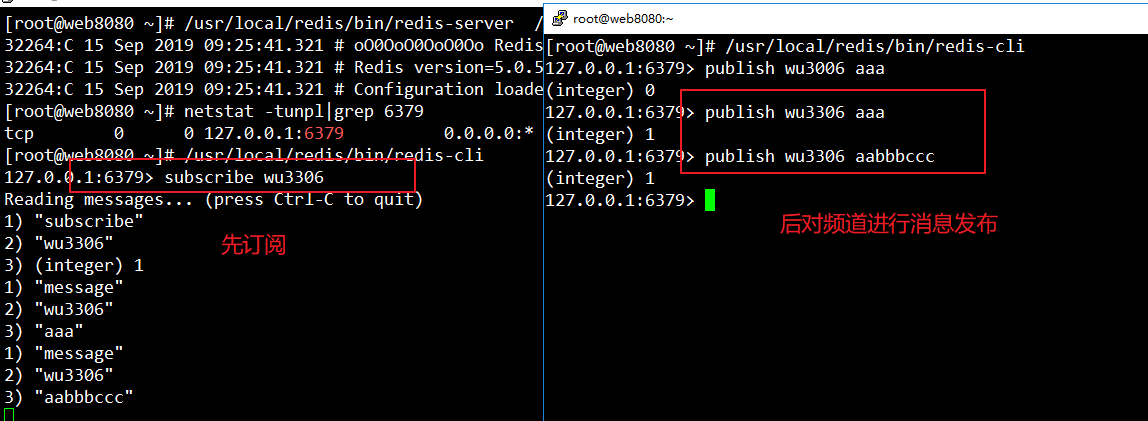
subscribe 频道1[,频道2…]

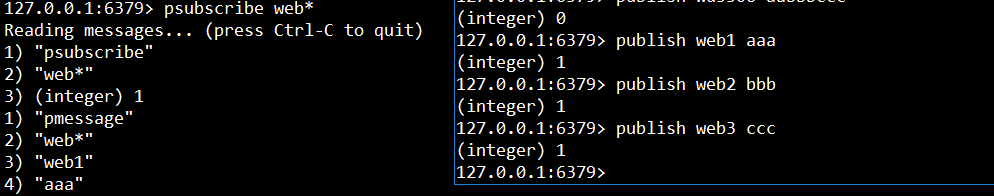
支持通配符

psubscribe 名称\*

发布频道

publish 频道 发送消息





# 二、数据持久化操作

数据持久化就是在服务重启或服务器重启后数据不丢失。实现持久化，就需要把数据存储到磁盘中。

Redis的持久化有2种方式

* 快照(rdb)默认就开启
* aof日志方式(需要手动开启)

两种的持久化的机制不相同，rdb在某一个时间点把内存中的数据整体保存下来。aof是把用户操作的命令全部记录下来。记录全部命令会对性能有一定的损耗，所以默认redis就没有开启，有条件化建议开启。

## 2.1、rdb快照

相关配置选项 vi /usr/local/redis/redis.conf

# 秒 命令次数

save 900 1 # 900秒内,有1条写入,则产生快照

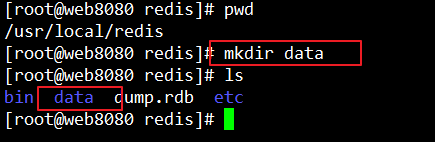
save 300 10 # 300秒内有10次写入,则产生快照

save 60 10000 # 60秒内有10000次写入,则产生快照

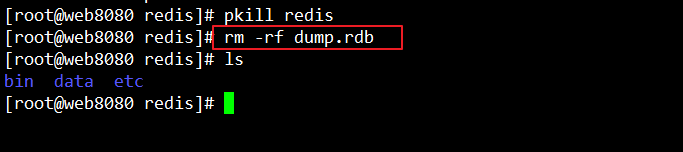
dbfilename dump.rdb //导出来的rdb的默认文件名

dir ./ // 持久化文件存放路径

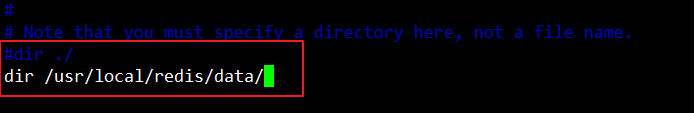
创建一个数据快照存储目录



关闭redis服务，再删除原来的dump.rdb



更改快照生成文件所在的目录路径



保存配置文件后，重启redis服务。

使用redis提供的压测命令来生成10001个key





## 2.2、aof日志持久化

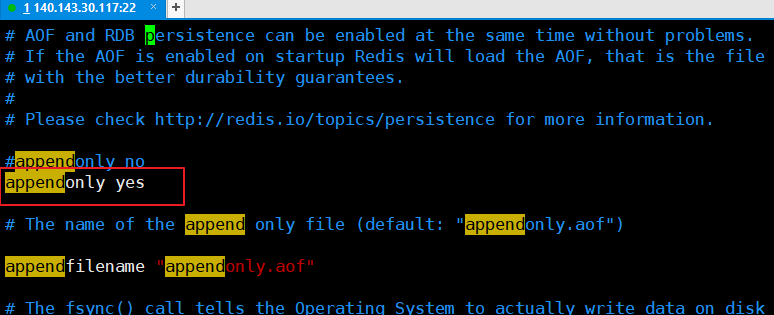
相关配置 vi /usr/local/redis/redis.conf

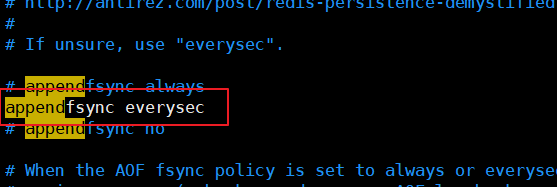
appendonly no # 是否打开 aof日志功能 no不打开 yes打开

#appendfsync always # 每1个命令,都立即同步到aof. 安全,速度慢

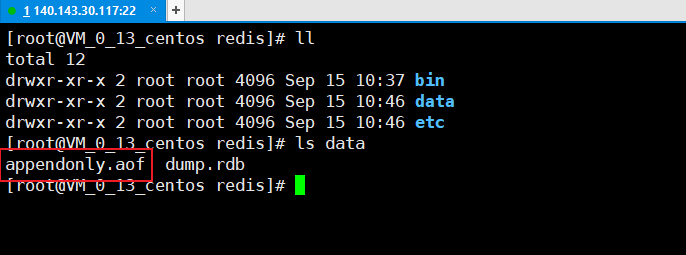
appendfsync everysec # 推荐方案,每秒写1次

#appendfsync no # 写入工作交给操作系统,数据同步性没有保证,同步频率低,速度快





进入到redis命令客户端中，执行几次set操作，就可以在指定的日志目录中查看到对应的日志文件



# 三、Redis中的事务

Redis支持简单的事务，事务就是：当同一个操作需要多条命令执行，一条执行有误，其它操作将回滚到之前的状态。

例如：银行转账工作，从一个账号扣款并在另一个账户增款，要么都执行，要么都不执行。

执行的步骤：

开始事务

命令入队

执行事务

Redis事务实现

watch key1 key2 # 监听key的变化

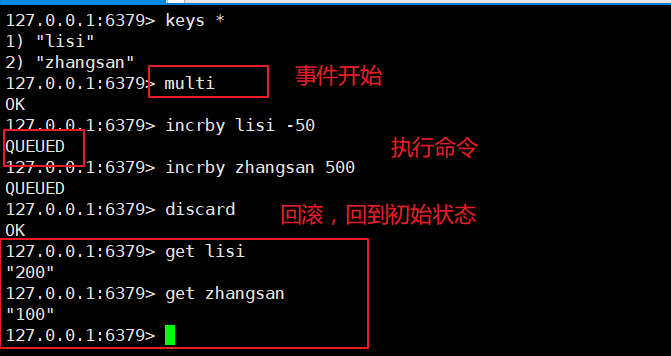
multi 事务开始

普通命令(string list hash set zset中的命令)

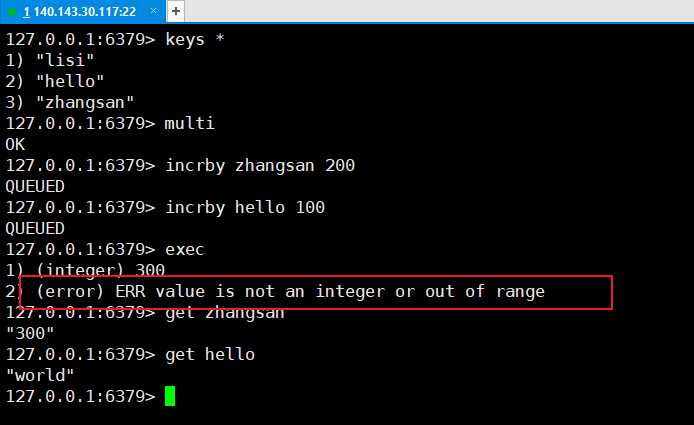
exec # 执行 / discard # 取消 exec和discard两个只能执行一个

unwatch # 解除监听

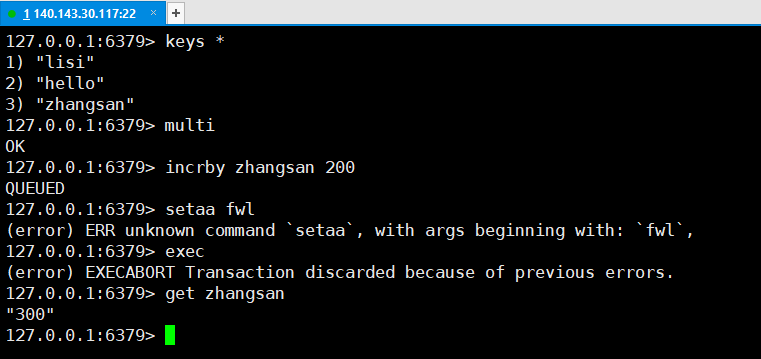
回滚命令 discard



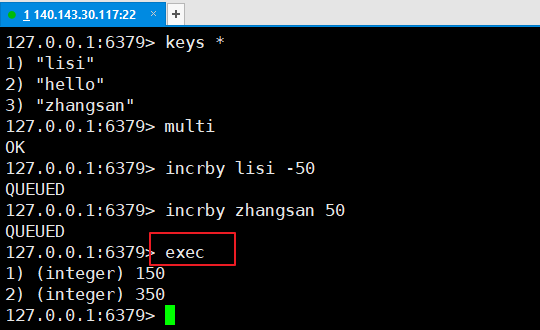
如果执行的命令没有错，只是业务有问题则不会自动回滚，会执行可以操作的队列中的命令



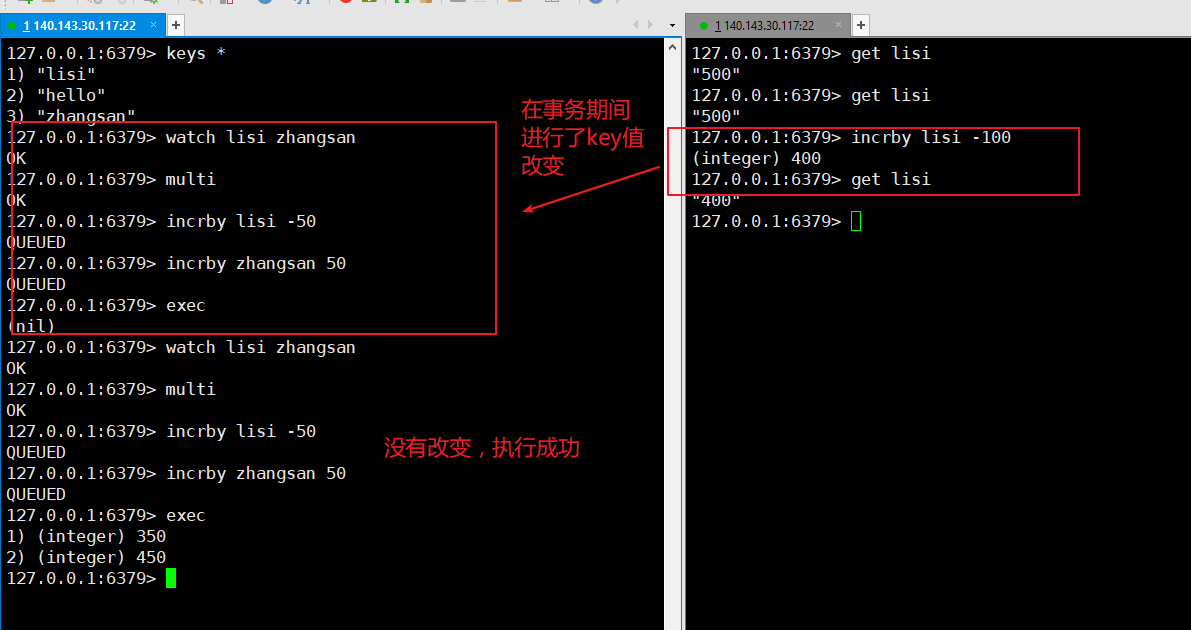
如果命令有错，则会自动回滚



事务执行操作



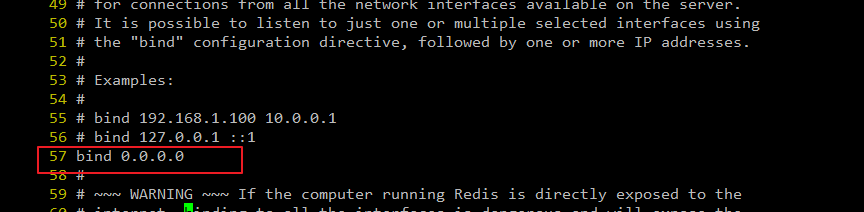
key的监听 watch 监听key是否有改变，如果有改变，则执行的事务将会不会成功执行。只有监听的在事务没有执行成功前没有改变，事务才能完成执行。



# 四、Redis密码安全

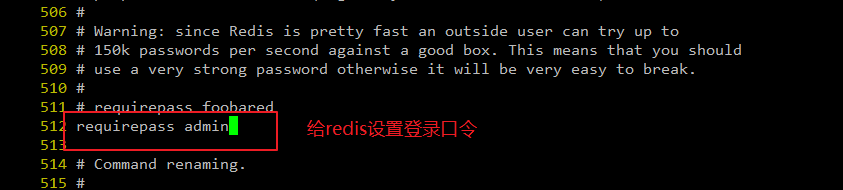
## 4.1、ip限制

vi /usr/local/redis/etc/redis.conf文件来通过配置文件限制ip访问,多个ip用空格隔开



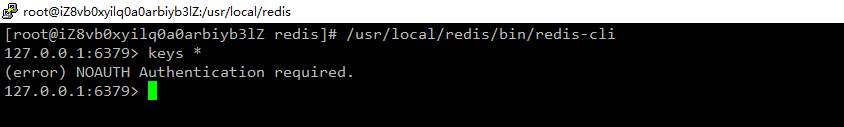
## 4.2、密码





重启redis服务

登录测试验证密码是否生效

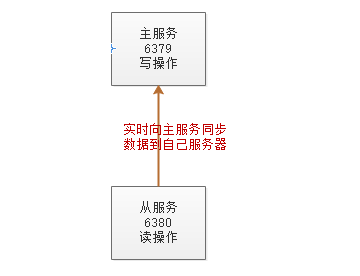
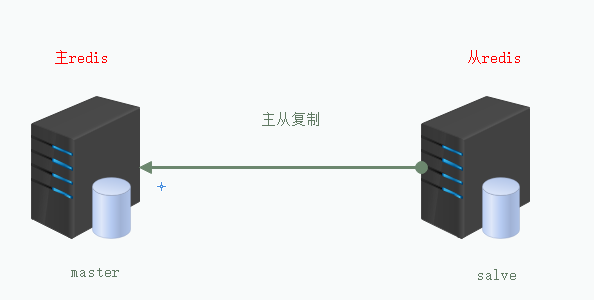


使用auth来进行登录操作



# 五、Redis主从设置

当数据量变得庞大的时候，读写分离还是很有必要的。redis提供了主从复制的机制，从服务器可以复制主服务器的数据信息，就可以实现读写分离，从而降低单台服务器的压力。

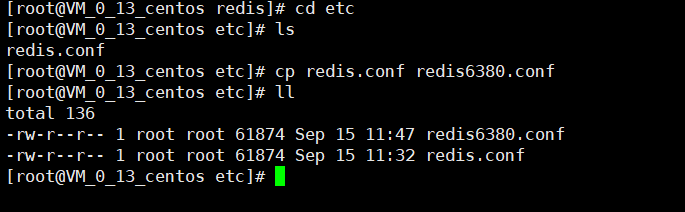


## 5.1、主服务器配置

开启rdb和aof日志记录,还有密码认证登录

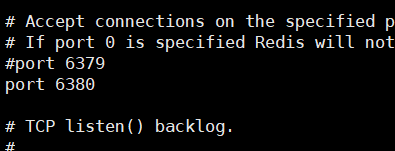
## 5.2、从服务器

因为是使用从端口，所以复制一个redis.conf文件

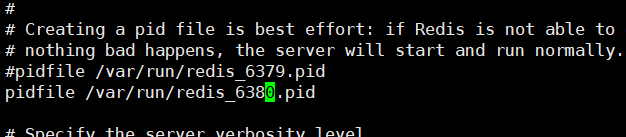


修改redis6380.conf文件中的相关记录配置

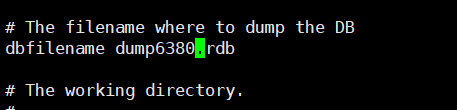
修改端口号



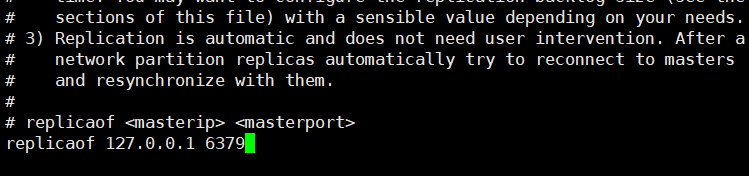
修改一下pid文件



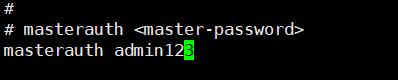
修改rdb文件名称



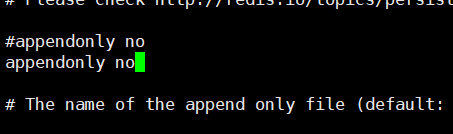
设置从服务器中连接主服务器的IP和端口号



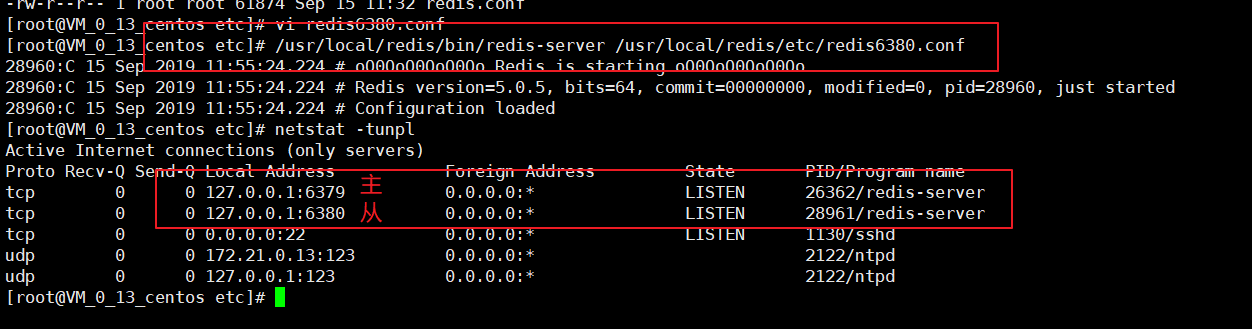
设置主服务器连接时的口令



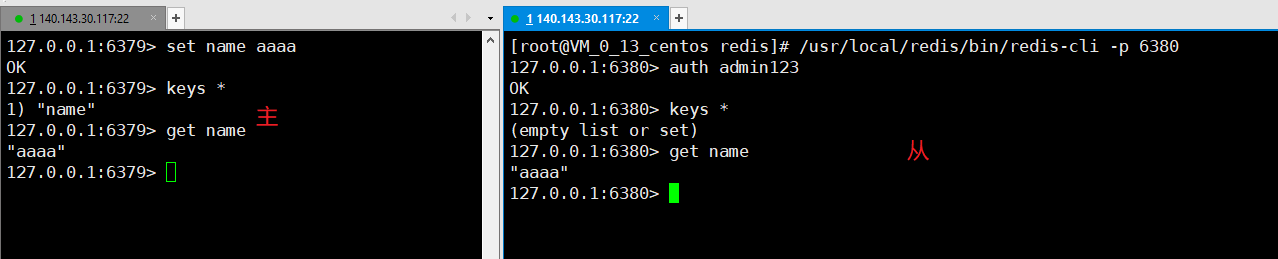
关闭aof日志记录



启动从服务



## 5.3、测试主从是否完成

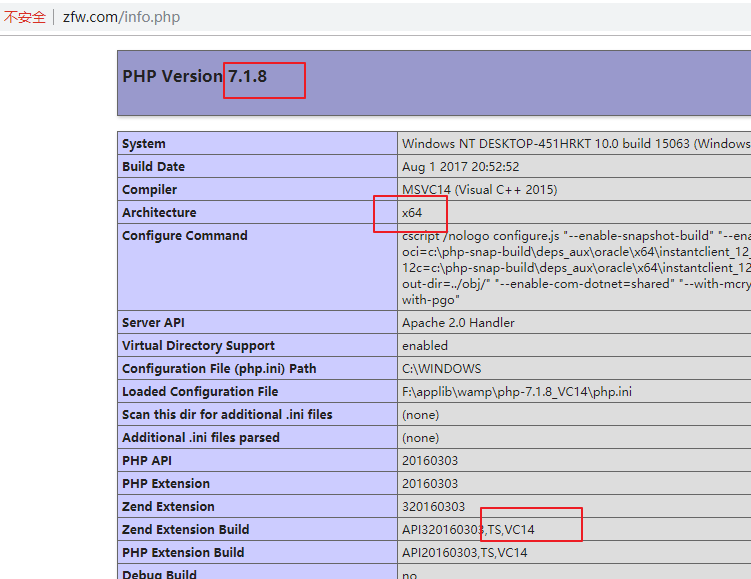


# 六、PHP操作Redis

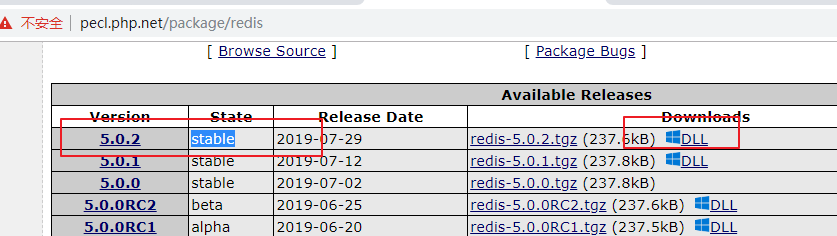
## 6.1、windows下安装扩展

下载地址：<https://pecl.php.net/package/redis>

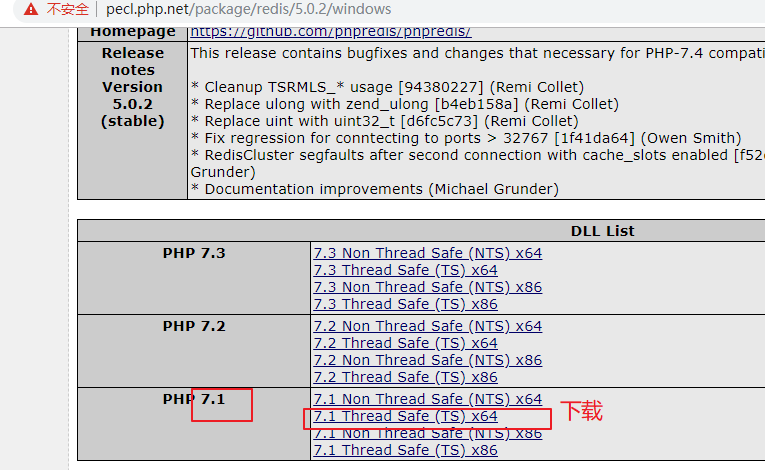
通过phpinfo函数查看当前php所支持的对应的版本



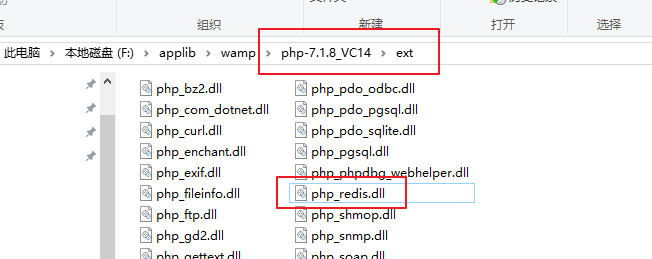
下载最新的windows支持的扩展文件



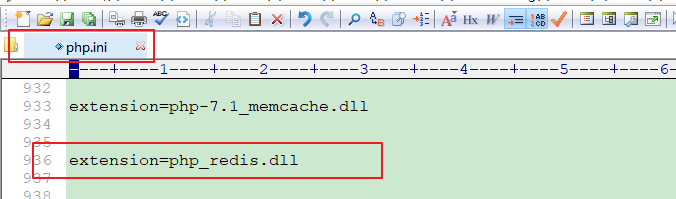
点击dll进入，下载对应phpinfo所显示支持的链接下载



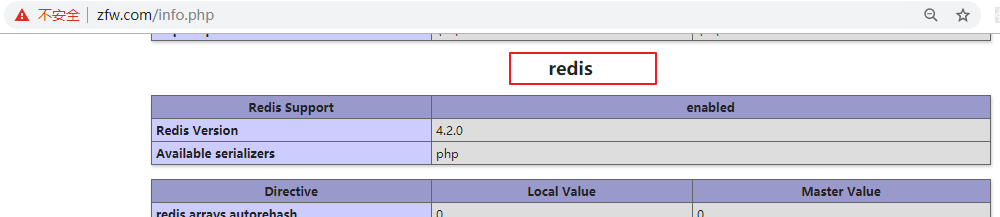
下载好后解压，复制dll文件到php安装目录中的ext目录下



修改php.ini文件，添加对应的扩展名称包括后缀名。

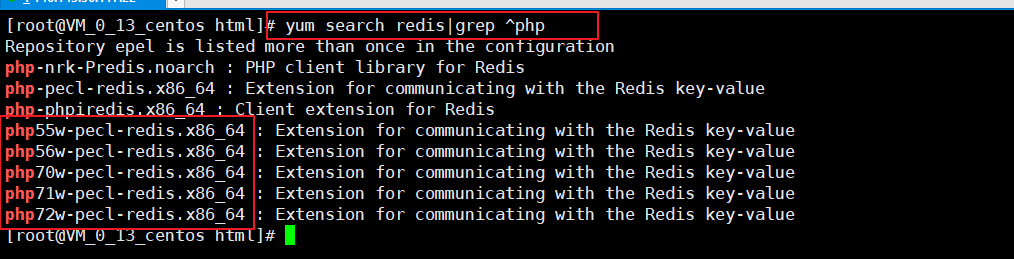


重启web服务器或php-fpm,通过phpinfo再次查看是否有redis字符串。



## 6.2、linux下安装redis对应php扩展

yum安装php所支持的扩展

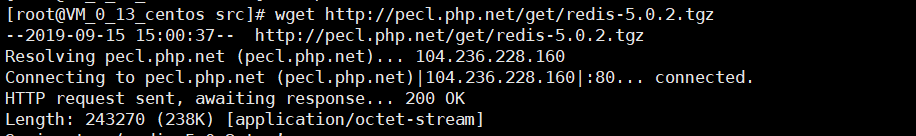


yum install -y php72w-pecl-redis.x86\_64

使用源码来安装，在php官网在下载对就的redis扩展源码

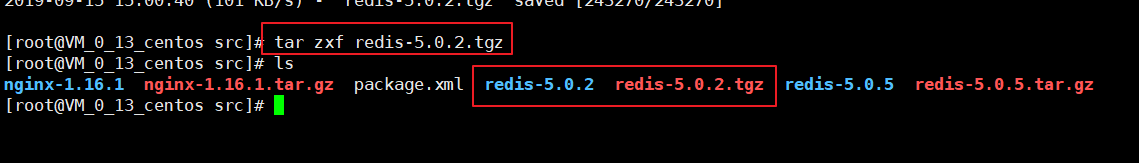
<http://pecl.php.net/package/redis>

wget http://pecl.php.net/get/redis-5.0.2.tgz

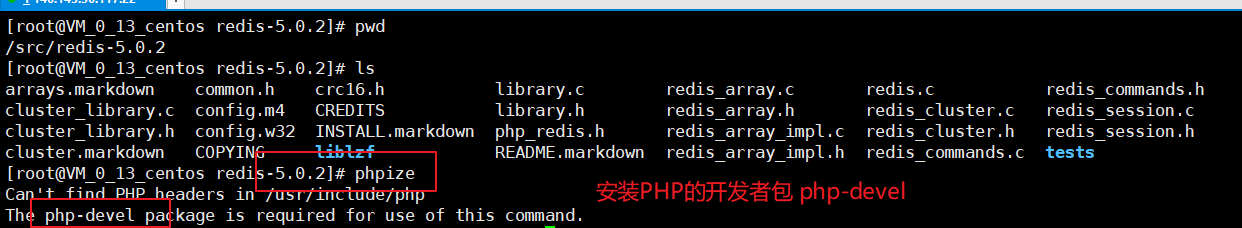


解压下载好的redis扩展源码包

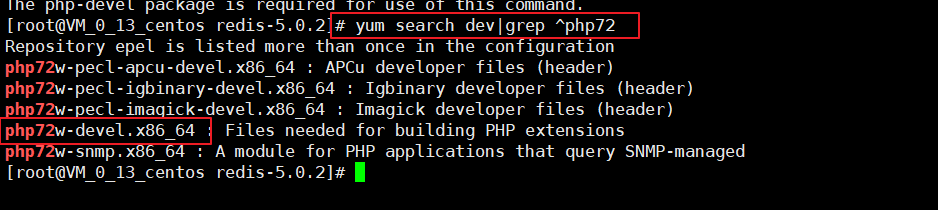
tar zxf redis-5.0.2.tgz



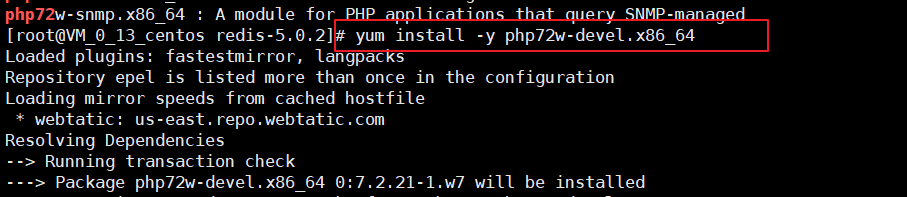
生成configure文件



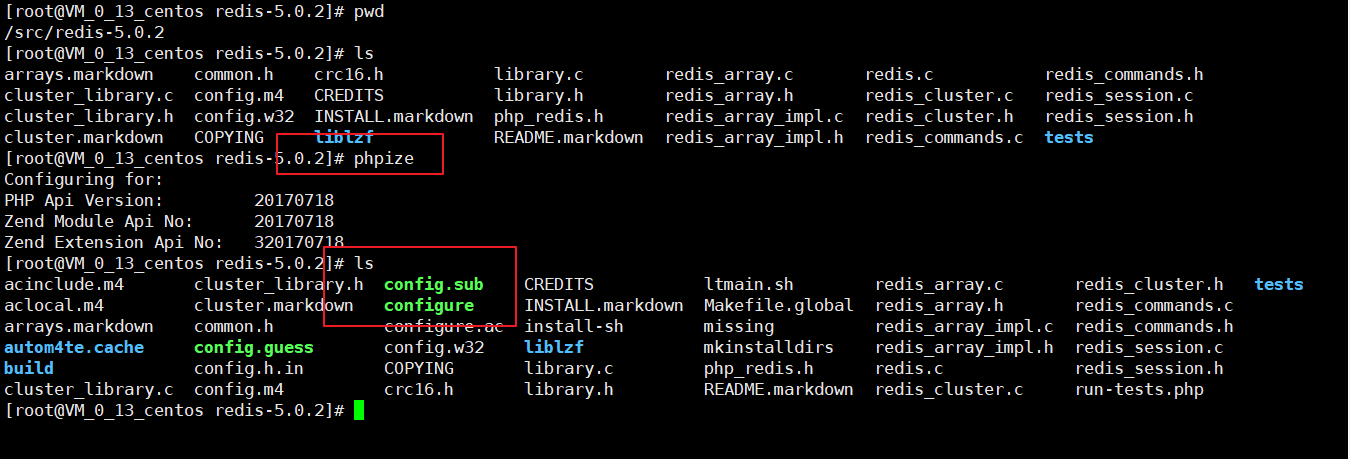
查找到对应PHP版本devel



yum install -y php72w-devel.x86\_64



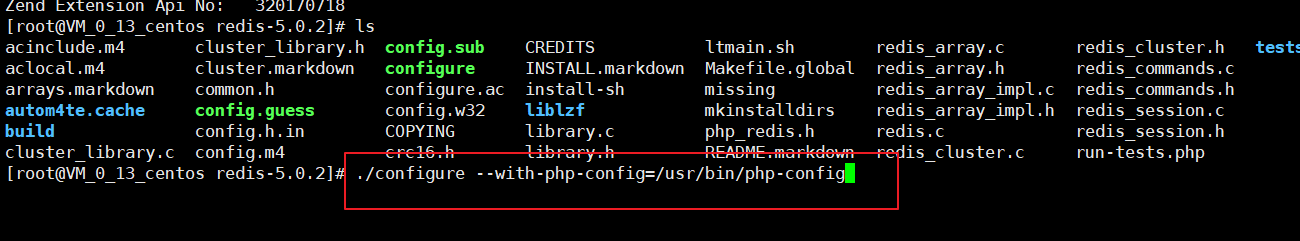
再次执行phpize



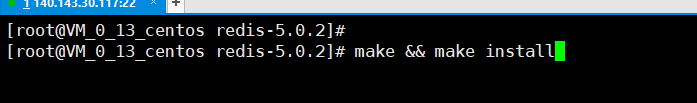
配置php扩展安装

./configure --with-php-config=/usr/bin/php-config

make && make install



检查配置没有报错，执行编译安装



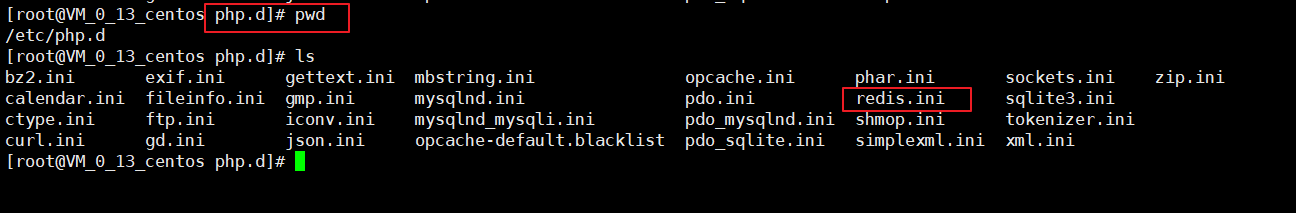
检查so文件中是否有redis的扩展文件

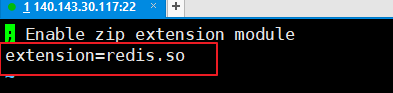


因为使用yum安装的php7.2版本，所以它推荐使用分文件来进行配置扩展。

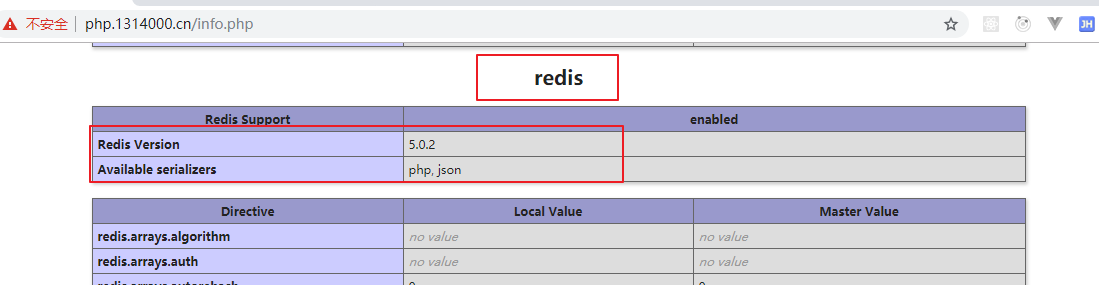
复制一个名为redis.ini文件

cp /etc/php.d/zip.ini /etc/php.d/redis.ini



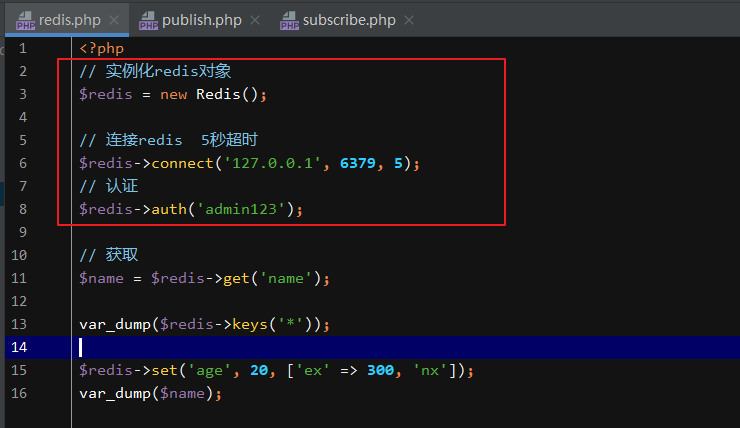


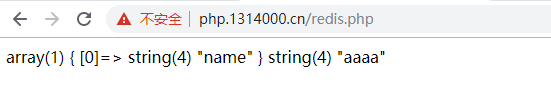
重启web服务器或php-fpm让扩展生效



## 6.3、php操作redis

在线手册：<https://github.com/phpredis/phpredis/#readme>





# 七、文章管理

## 7.1、项目需求

后台：

用户登录，登录成功后要有发送邮件功能(异常发送)

文章管理

文章列表、 添加文章、 修改文章、 删除文章

前台

热门文章推荐

数据保存的数据库为 redis

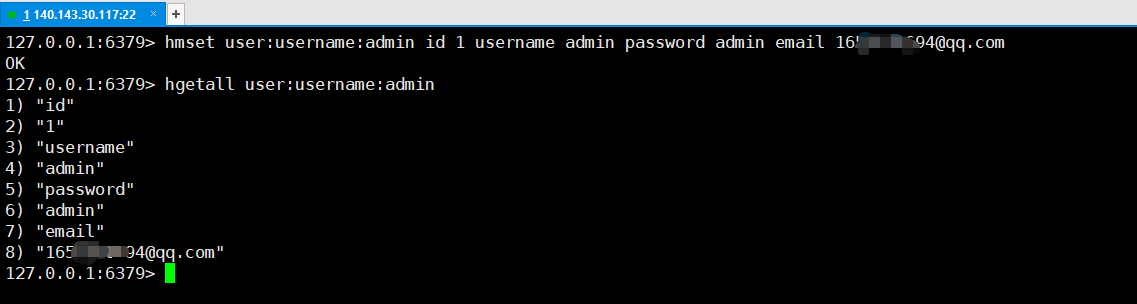
## 7.2、用户登录与邮件发送

用户表key设计

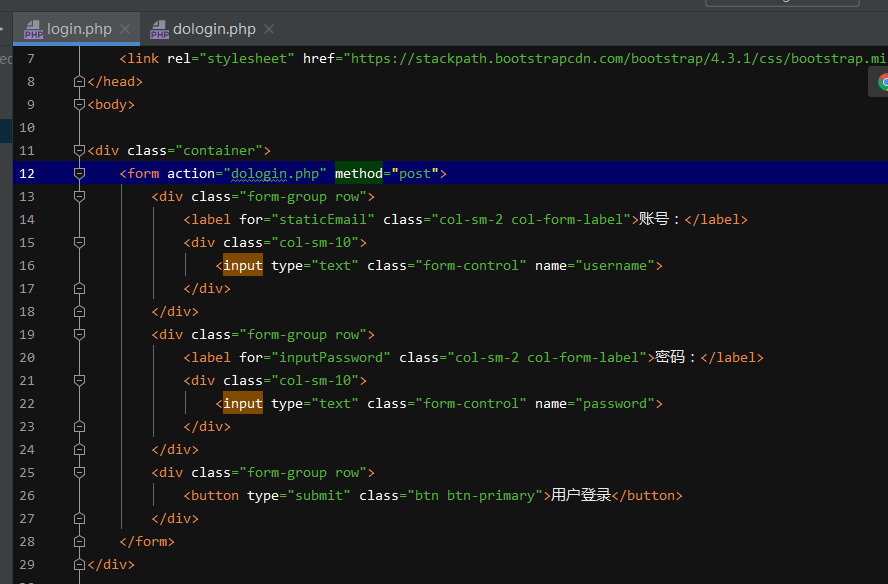
账号、密码

用hash类型 user:username:admin 字段 id:1 username:admin password:admin888

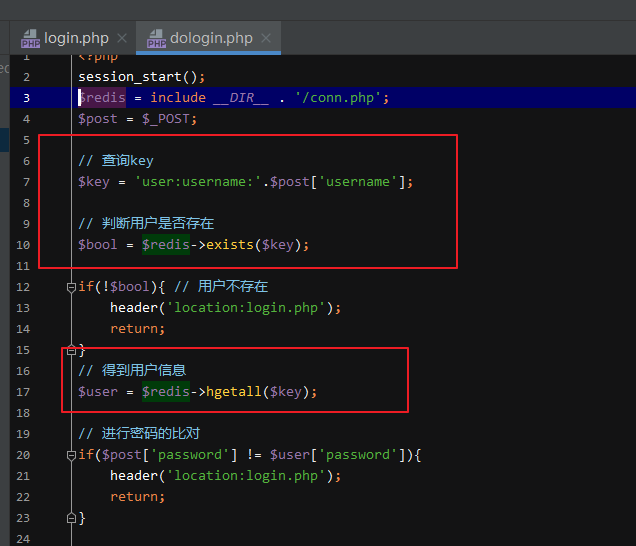
添加一条用户记录



登录和与任务到队列中

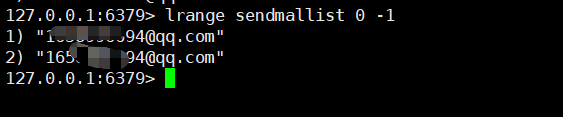


登录接受消息处理



发送邮件任务写入到队列中





## 7.3、设计文章key

列表 zset crc32

记录 hash

id string

article:id string key

article:id:1 字段 article:id:2 字段 hash key

以ID来排序

article:zset:id 1=>1 2=>2 zset key

## 7.4、文章列表

## 7.5、消费队列发邮件

安装发送邮件类库，phpmailer

composer require phpmailer/phpmailer

