1. Redis入门简介：

Redis是一个开源的使用ANSIC语言编写、支持网络、可**基于内存亦可持久化的日志型**、Key-Value数据库，并提供多种语言的API。

Redis是一个key-value存储系统。 和Memcached缓存类似，Redis支持存储的value类型相对更多，包括**string(字符串)**、**list(链表)**、**set(集合)**、**hash（哈希类型）**。并且支持在服务器端计算集合的并，交和补集(difference)等，还支持多种排序功能。Redis也被看成是一个数据结构服务器。

Redis支持主从同步，数据可以从主服务器向任意数量的从服务器上同步，从服务器可以是关联其他从服务器的主服务器。

2. 安装redis：

- yum安装:
- 源码编译安装：

```
# yum 安装：
yum  install redis -y
```

```
# 源码安装：
cd /usr/src
wget http://download.redis.io/redis-stable.tar.gz
tar xf redis-stable.tar.gz
cd redis-stable
# 编译安装redis
make PREFIX=/usr/local/redis install
cp redis.conf /usr/local/redis/6379.conf
# 到这一步，redis已经安装好了
# 在前端运行redis:
/usr/local/redis/bin/redis-server
/usr/local/redis/6379.conf
# 后台运行redis：
nohup  /usr/local/redis/bin/redis-server
/usr/local/redis/6379.conf  &
```

```
#关闭redis服务：
/usr/local/redis/bin/redis-cli -p 6379 shutdown

# 启动之后，redis日志可能会出现一些警告信息：
1629:M 24 Nov 01:50:41.918 # WARNING: The TCP backlog
setting of 511 cannot be enforced because
/proc/sys/net/core/somaxconn is set to the lower value of
128.
1629:M 24 Nov 01:50:41.918 # Server started, Redis version
3.2.12

1629:M 24 Nov 01:50:41.918 # WARNING overcommit_memory is
set to 0! Background save may fail under low memory
condition. To fix this issue add
'vm.overcommit_memory = 1' to /etc/sysctl.conf and then
reboot or run the command 'sysctl vm.overcommit_memory=1'
for this to take effect.

1629:M 24 Nov 01:50:41.918 # WARNING you have Transparent
Huge Pages (THP) support enabled in your kernel. This will
create latency and memory usage issues with Redis. To fix
this issue run the command
'echo never > /sys/kernel/mm/transparent_hugepage/enabled'
as root, and add it to your /etc/rc.local in order to
retain the setting after a reboot. Redis must be restarted
after THP is disabled.

#第一个报错是backlog参数导致的，backlog主要控制的是三次握手的时候，
server端收到客户端ack确认号的队列长度，如果队列满了，客户端将收到错
误提示。
vim /etc/sysctl.conf
net.core.somaxconn=512

#第二个报错是因为vm.overcommit_memory=0,0表示内核会检查有没有多的
内存给进程使用，
如果没有，内存的申请失败，并将错误返回给进程。可以设置为1，表示内核允
许分配所有的物理内存给进程。
还有2，表示所有的物理内存和swap全部给进程。
vim /etc/sysctl.conf
vm.overcommit_memory=1
```

```
#第三个报错是开启了大页内存动态分配，设置关闭，让redis负责内存管理
echo never > /sys/kernel/mm/transparent_hugepage/enabled

# 然后执行一下命令，重启redis就不会提示警告信息了：
sysctl -p

# 这样启动有点麻烦，也可以配置脚本启动：
/usr/src/redis-stable/utils/install_server.sh
Welcome to the redis service installer
This script will help you easily set up a running redis
server

#配置redis的端口，可以默认为6379，直接回车
Please select the redis port for this instance: [6379]
Selecting default: 6379
#配置redis的配置文件，放到redis的安装目录，加上文件名：
Please select the redis config file name
[/etc/redis/6379.conf] /usr/local/redis/6379.conf
#配置redis的日志，放到redis的安装目录，加上文件名：
Please select the redis log file name
[/var/log/redis_6379.log] /usr/local/redis/redis_6379.log
#配置redis的数据目录，也放在安装目录，加上端口号区分：
Please select the data directory for this instance
[/var/lib/redis/6379] /usr/local/redis/6379
#选择redis的启动程序，加上文件名
Please select the redis executable path []
/usr/local/redis/bin/redis-server
Selected config:
Port           : 6379
Config file    : /usr/local/redis/6379.conf
Log file       : /usr/local/redis/redis_6379.log
Data dir       : /usr/local/redis/6379
Executable     : /usr/local/redis/bin/redis-server
Cli Executable : /usr/local/redis/bin/redis-cli
Is this ok? Then press ENTER to go on or Ctrl-C to abort.
Copied /tmp/6379.conf => /etc/init.d/redis_6379
Installing service...
Successfully added to chkconfig!
Successfully added to runlevels 345!
Starting Redis server...
Installation successful!
```

3. lnmp整合redis：

已经部署好lnmp，并发布了一套门户网站。

3.1 安装php-redis连接驱动：

现在redis安装好了，但是还是没法缓存数据，因为php还没有于redis建立联系：

```
# 源码方式安装php，整合redis：
wget
https://github.com/phpredis/phpredis/archive/4.2.0.tar.gz
tar xf 4.2.0.tar.gz
cd phpredis-4.2.0/
/usr/local/php/bin/phpize
./configure  --with-php-config=/usr/local/php/bin/php-
config  --enable-redis
make && make install

#编辑php.ini配置文件，添加redis模块：
extension=redis.so

# yum方式安装php，整合redis：
安装php7.0
yum install https://rpms.remirepo.net/enterprise/remi-
release-7.rpm -y
yum install yum-utils -y
yum-config-manager --enable remi-php70
yum install php php-fpm php-mysql php-redis -y
```

3.2 测试redis模块是否添加成功：

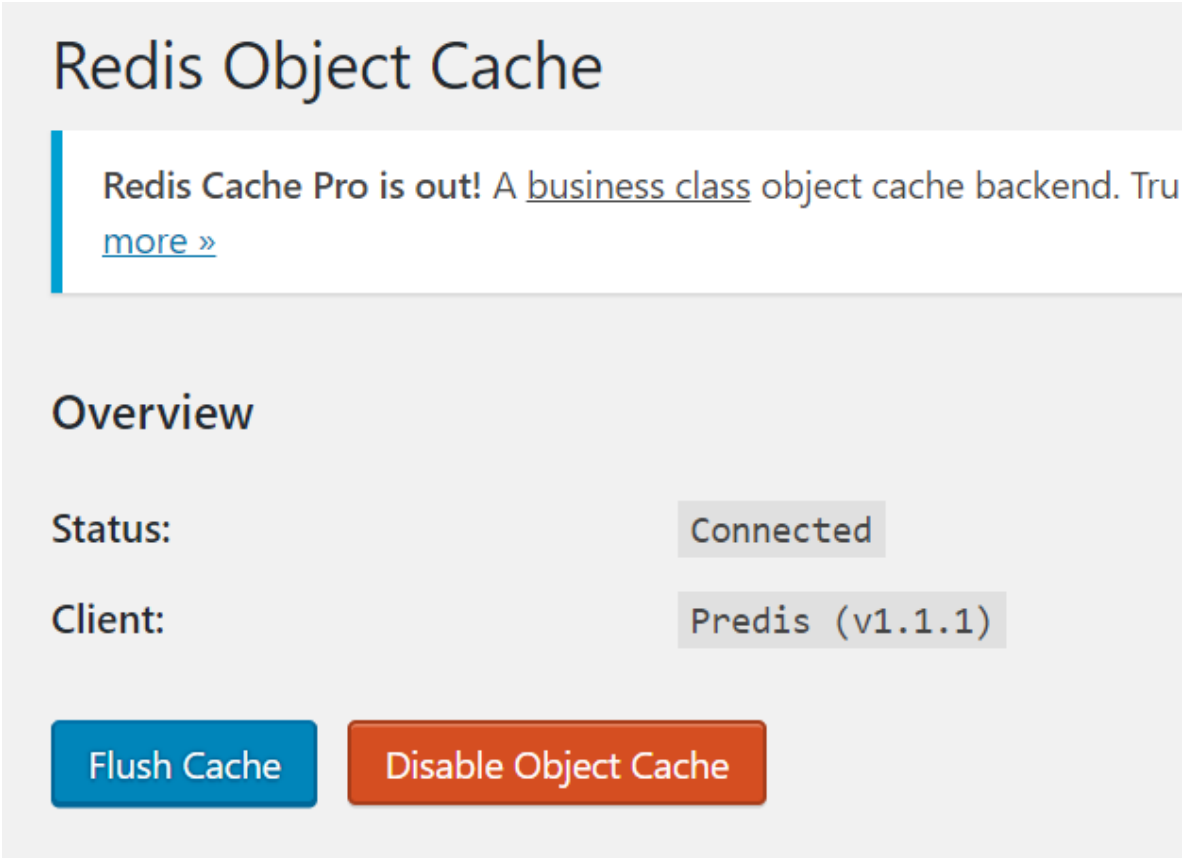
```
vim  /usr/local/nginx/html/wordpress/info.php

<?php
        phpinfo();
?>
```

访问，可看到redis模块即可：

3.3 安装redis插件：

在wordpress后台，安装Redis Object Cache插件，并且启用，看到如果已连接，表示成功：



3.4 查看缓存信息：

也可以在shell中通过客户端工具连接redis服务端，查看缓存信息：

```
[root@localhost ~]# /usr/local/redis/bin/redis-cli
127.0.0.1:6379> keys *
 1) "wp:comment:get_comments-
b8d07687c3930bd5a8a032368044cf67-0.07470500 1573122008"
 2) "wp:posts:1"
 3) "wp:terms:1"
```

3.5 配置远程redis服务器：

如果redis是在远程服务器上，则需要修改redis服务器所在的ip地址：

```
vim /usr/local/nginx/html/wordpress/wp-content/object-
cache.php

$parameters = array(
            'scheme' => 'tcp',
            'host' => '192.168.75.134',
```

```
            'port' => 6379,
            'timeout' => 5,
            'read_timeout' => 5,
            'retry_interval' => null
        );

# 如果配置虚拟主机，还有一个discuz网站也需要缓存，
如果不加以配置，两个站点数据都会缓存在一个数据库中，
不太便于管理，可以让不同应用选择不同的数据库进行缓存。

# 修改discuz的网站配置文件，指定数据库：
vim  /usr/local/nginx/html/upload/config/config_global.php

$_config['memory']['redis']['serializer'] = 1;
$_config['memory']['redis']['db'] = 2;


# 进一步修改discuz的配置文件：
vim
/usr/local/nginx/html/upload/source/class/memory/memory_dr
iver_redis.php

@$this->obj->setOption(Redis::OPT_SERIALIZER,
$config['serializer']);
                                $this-
>select($config['db']);
```

3.6 正式解读redis配置文件：

```
# 监听地址，默认是监听本地回环地址，可以修改为0.0.0.0，表示监听所有
ip地址，如果要监听多个，用空格隔开。
bind 127.0.0.1
# 开启保护模式，如果没有配置bind，而是直接注释了bind，那么远程服务器
是不能直接查看键值数据的，可以设置为no（不建议），或者设置bind监听端
口，不要直接注释;再或者设置密码requirepass 123456
protected-mode yes
# 监听端口，可以修改
port 6379
# ack队列长度
tcp-backlog 511
# 客户端与服务端的连接超时时间，0表示永不超时
```

```
timeout 0
会话探测时间，redis服务端默认每隔300s发ack包给客户端，探测客户端是否还在，还在就保持连接
tcp-keepalive 300
# 默认情况下redis是在前台运行，可以设置为yes,让redis以守护进程在后台运行。
daemonize no
supervised no
pidfile /var/run/redis_6379.pid
# 日志级别，有debug,verbose,notice,warning
loglevel notice
# 日志路径
logfile /usr/local/redis/6379.log
# 是否把日志输出到系统日志，默认为no
syslog-enabled no
# 设置数据库个数，从0号数据库开始，默认为17个
databases 16
# 在启动时是否显示日志
always-show-logo yes
# 在900秒内修改一个键触发快照
save 900 1
# 在300秒修改10个键触发快照
save 300 10
# 在60秒修改10000个键触发快照
save 60 10000
# 在快照出现问题时，禁止redis写入操作
stop-writes-on-bgsave-error yes
# 进行持久化时，是否压缩，默认为压缩
rdbcompression yes
# 在保存或者加载rdb数据库时是否开启校验
rdbchecksum yes
# rdb文件名，可以修改
dbfilename dump.rdb
# 数据库文件存放路径
dir /usr/local/redis/6379
```

4. redis备份概念：

Redis所有数据都是保存在内存中，Redis数据备份可以定期的通过异步方式保存到磁盘上，该方式称为半持久化模式，如果每一次数据变化都写入aof文件里面，则称为全持久化模式。同时还可以基于Redis主从复制实现Redis备份与恢复。

5. redis备份模式：

- 半持久化rdb模式
- 全持久化aof模式
- redis主从复制

5.1 半持久化RDB模式

半持久化**RDB模式是Redis备份默认方式**，是通过快照（snapshotting）完成的，当符合在Redis.conf配置文件中设置的条件时Redis会自动将内存中的所有数据进行快照并存储在硬盘上，完成数据备份。

Redis启动后会读取RDB快照文件，将数据从硬盘载入到内存，根据数据量大小与结构和服务器性能不同，通常将一个记录一千万个字符串类型键、大小为1GB的快照文件载入到内存中需花费20～30秒钟。

rdb持久化实现过程：

Redis实现快照的过程，Redis使用fork函数复制（写时复制）一份当前进程（父进程）的副本（子进程），父进程继续接收并处理客户端发来的命令，而子进程开始将内存中的数据写入硬盘中的临时文件，当子进程**写入完所有数据**后会用该临时文件替换旧的RDB文件，至此一次快照操作完成。

开启rdb持久化：

Redis进行RDB快照的条件由用户在配置文件中自定义，由两个参数构成：**时间和改动的键的个数。**

当在指定的时间内被更改的键的个数大于指定的数值时就会进行快照。在配置文件中已经预置了3个条件：

```
save 900 1 #900秒内有至少1个键被更改则进行快照；
save 300 10 #300秒内有至少10个键被更改则进行快照；
save 60 10000 #60秒内有至少10000个键被更改则进行快照
```

默认可以存在多个条件，条件之间是“或”的关系，只要满足其中一个条件，就会进行快照。

禁用rdb持久化：

如果想要禁用自动快照，只需要将所有的save参数删除（注释）即可。

rdb数据文件路径：

Redis默认会将快照文件存储在Redis数据目录，默认文件名为：dump.rdb文件，可以通过配置dir和dbfilename两个参数分别指定快照文件的存储路径和文件名。也可以在Redis命令行执行`config get dir`获取Redis数据保存路径。

RDB文件是经过压缩（可以配置rdbcompression参数以禁用压缩节省CPU占用）的二进制格式，所以占用的空间会小于内存中的数据大小，更加利于传输。

除了自动快照，还可以**手动发送SAVE和BGSAVE命令让Redis执行快照**，两个命令的区别在于：

- save是由主进程进行快照操作，会阻塞住其他请求。
- bgsave是通过fork子进程进行快照操作。

rdb持久化常用参数：

```
save 900 1          900秒内有1个键发生变化开始快照
save 300 10         300秒内有10个键发生变化开始快照
save 60 10000       60秒内有10000个键发生变化开始快照
# 快速理解，可以从下往上去看，即使只有一个键发生变化，最后也会执行900秒的规则。

#后台存储发生故障时，客户端停止写入
stop-writes-on-bgsave-error yes
```

```
#在存储过程中，启动压缩
rdbcompression yes

#启动redis时，是否检查rdb数据库的完整性
rdbchecksum yes

#rdb数据库的名字，可自定义
dbfilename dump.rdb

#rdb数据库存放路径，可以自定义
dir /var/lib/redis
```

5.2 全持久化AOF模式：

如果数据很重要无法承受任何损失，可以考虑使用AOF方式进行持久化，默认Redis没有开启AOF(append only file)方式的全持久化模式。

aof持久化实现过程：

开启AOF持久化后每执行一条会更改Redis中的数据的命令，Redis就会将该命令写入硬盘中的AOF文件。

在启动时Redis会逐个执行AOF文件中的命令来将硬盘中的数据载入到内存中，载入的速度相较RDB会慢一些。

开启aof持久化：

redis默认是没有开启全持久化的，需要在配置文件修改参数实现：

```
#开启aof持久化，默认为禁止no
appendonly yes
```

aof数据文件路径：

AOF文件的保存位置和RDB文件的位置相同，都是通过dir参数设置的，默认的文件名是appendonly.aof，可以通过appendfilename参数修改该名称。

Redis允许同时开启AOF和RDB，既保证了数据安全又使得进行备份等操作十分容易。

aof持久化常用参数：

```
#开启aof持久化，默认为禁止no
appendonly yes

#指定aof文件名，可自定义
appendfilename "appendonly.aof"

#每执行一条命令即写入磁盘
#appendfsync always

#每秒同步内存数据到磁盘
appendfsync everysec

#设置为no,让写入动作交由操作系统完成，同步的频率会相对比较低，不建议
# appendfsync no

#在rdb写磁盘过程中，是否需要停止aof，默认是不停，如果负载较高，建议停止。
# 这个时候数据也不会丢失，在暂时存在内存队列，rdb数据写完后，aof继续。
no-appendfsync-on-rewrite no

#aof文件大小相比上次文件大小，增长100%时，重写
auto-aof-rewrite-percentage 100

#aof文件至少超过64M时，重写
auto-aof-rewrite-min-size 64mb
# redis在恢复时，会忽略最后一条可能存在问题的指令（因为断电等原因导致的一些错误指令）
aof-load-truncated yes
```