**NOSQL-day01**

# NOSQL数据库整理——**搭建Redis数据库服务器**

## 部署Redis服务

### 数据库类型

**RDBMS**——关系数据库管理系统

Relational Database Management System

按照预先设置的组织结构，将数据存储在物理介质上

数据之间可以做关联操作

**RDBMS服务软件**（主流的RDBMS软件有以下几种）

Oracle

DB2

MS SQL Server

MySQL、MariaDB（开源跨平台）

**NOSQL**（NOSQL=Not Only SQL）

— 意思是“不仅仅是SQL”

— 泛指非关系型数据库

— 不需要预先定义数据存储结构

— 表的每条记录都可以有不同的类型和结构

**NOSQL服务软件**（主流软件）

Redis MongoDB Memcached CouchDB Neo4j FlockDB

### Redis介绍（www.redis.cn）

— Remote Dictionary Server （远程字典服务器）

— 是一款高性能的（Key/Values）分布式内存数据库

— 支持数据持久化，可以把内存里的数据保存到硬盘中

— 也支持list、hash、set、zset数据类型

**配置步骤：**

### 1. 装包

[root@client50 ~]# rpm -q gcc //查询依赖包是否安装

[root@client50 ~]# yum -y install gcc //安装依赖包

[root@client50 ~]# tar -xf redis-4.0.8.tar.gz

**源码编译安装：**

[root@client50 ~]# cd redis-4.0.8/

[root@client50 redis-4.0.8]# make && make install

**初始化配置：**

[root@client50 redis-4.0.8]# ./utils/install\_server.sh //使用默认即可

Port : 6379 //端口

Config file : /etc/redis/6379.conf //主配置文件

Log file : /var/log/redis\_6379.log //数据库日志文件

Data dir : /var/lib/redis/6379 //pid文件

Executable : /usr/local/bin/redis-server //启动程序

Cli Executable : /usr/local/bin/redis-cli

### 2. 服务管理

**启动服务**

[root@client50 redis-4.0.8]# /etc/init.d/redis\_6379 status

**停止服务**

[root@client50 redis-4.0.8]# /etc/init.d/redis\_6379 stop

**查看状态**

[root@client50 redis-4.0.8]# /etc/init.d/redis\_6379 start

### 3. 连接Redis数据库服务

**访问redis服务**

[root@client50 ~]# ps -C redis-server

[root@client50 ~]# ss -natpul | grep redis-server

**本机访问数据库**

[root@client50 ~]# redis-cli

127.0.0.1:6379> keys \* //打印所有变量

(empty list or set)

127.0.0.1:6379> exit

**常用操作指令**

|  |  |  |
| --- | --- | --- |
| 指令 | 定义 | 示例 |
| set keyname keyvalue | 存储 | :6379[1]> set name lucy |
| get keyname | 获取 | :6379[1]> get name |
| select 数据库编号0-15 | 切换库 | :6379> select 1 |
| keys \* | 打印所有变量 | :6379[1]> keys \* |
| keys a? | 打印指定变量 | :6379[1]> keys a? |
| EXISTS keyname | 测试变量是否存在 | :6379[1]> exists name  1为存在，0为不存在 |
| persist key | 设置key用不过期 |  |
| ttl keyname | 查看生存时间 | :6379[1]> ttl name  -1永久存储，-2生存时间归零 |
| type keyname | 查看类型 | :6379[1]> type name |
| move keyname dbname | 移动变量 | [1]> set name 1  [1]> move age 2  [2]> keys \* |
| expire keyname 10 | 设置有效时间 | :6379[1]> expire age 21 |
| del keyname | 删除变量 | [1]> del id |
| flushall | 删除所有变量 | [1]> flushall |
| save | 即刻保存所有变量到硬盘 | [1]> save |
| shutdown | 关闭redis服务，同时执行存盘操作 | [1]> shutdown |

### 4. 配置文件解析

[root@client50 ~]# vim /etc/redis/6379.conf

**数据单位：**

12 # 1k => 1000 bytes

13 # 1kb => 1024 bytes

14 # 1m => 1000000 bytes

15 # 1mb => 1024\*1024 bytes

16 # 1g => 1000000000 bytes

17 # 1gb => 1024\*1024\*1024 bytes

**常用配置选项：**

70 bind 127.0.0.1 //ip地址

93 port 6379 //端口

端口范围0～65535，可以修改，但不能使用已经被占用的端口

102 tcp-backlog 511 //tcp连接总数

114 timeout 0 //连接超时时间

131 tcp-keepalive 300 //长连接时间（S）

每隔300秒发送连接确认包，若回复则保持继续连接，若不回复，则断开

137 daemonize yes //守护进程方式运行

172 logfile /var/log/redis\_6379.log //pid文件

187 databases 16 //数据库个数

254 dbfilename dump.rdb //存储内存里的数据文件（数据库目录下）

264 dir /var/lib/redis/6379 //数据库目录

533 # maxclients 10000 //并发连接数量

565 # volatile-lru //最近最少使用（针对设置了TTL的key）

566 # allkeys-lru //删除最少使用的key

569 # volatile-random //在设置了TTL的key里随机移除

570 # allkeys-random //随机

571 # volatile-ttl（minor TTL） //移除最近过期的key

572 # noeviction //不删除，写满时报错

**内存管理：**

560 # maxmemory <bytes> //最大内存

591 # maxmemory-policy noeviction //定义使用策略

602 # maxmemory-samples 5 //选取模版数据的个数（针对lru和ttl策略）

**更改端口和ip地址之后，连接数据库时需要指定IP和端口号**

[root@client50 ~]# vim /etc/redis/6379.conf

70 bind 192.168.4.50 //ip地址

93 port 6350

[root@client50 ~]# /etc/init.d/redis\_6379 start

[root@client50 ~]# redis-cli -h 192.168.4.50 -p 6350

停止服务

[root@client50 ~]# redis-cli -h 192.168.4.50 -p 6350 shutdown

修改脚本之后使用默认方式停止服务

[root@client50 ~]# vim /etc/init.d/redis\_6379

43 $CLIEXEC -p 6350 -h 192.168.4.50 shutdown

[root@client50 ~]# /etc/init.d/redis\_6379 stop

### 4. 设置连接密码：

[root@client50 ~]# vim /etc/redis/6379.conf

501 requirepass 123456

[root@client50 ~]# /etc/init.d/redis\_6379 restart

[root@client50 ~]# redis-cli -h 192.168.4.50 -p 6350 //连接服务

192.168.4.50:6350> ping //会报错error

192.168.4.50:6350> auth 123456 //输入连接密码

192.168.4.50:6350> ping //正确输出PONG

**连接时直接加选项输入密码连接**

[root@client50 ~]# redis-cli -h 192.168.4.50 -p 6350 -a 123456

**停止服务**

[root@client50 ~]# /etc/init.d/redis\_6379 stop //停止服务失败

[root@client50 ~]# redis-cli -h 192.168.4.50 -p 6350 -a 123456 shutdown //加密码停服务

**修改脚本内容使用默认程序停止服务**

[root@client50 ~]# vim /etc/init.d/redis\_6379

43 $CLIEXEC -p 6350 -h 192.168.4.50 -a 123456 shutdown

[root@client50 ~]# /etc/init.d/redis\_6379 start

[root@client50 ~]# ss -natpul | grep redis

## 部署LNMP+Redis

### 部署LNMP

**安装源码nginx和php**

[root@web56 ~]# yum -y install gcc pcre-devel

[root@web56 ~]# tar -zxvf nginx-1.12.2.tar.gz

[root@web56 ~]# cd nginx-1.12.2/

[root@web56 ~]# ./configure --prefix=/usr/local/nginx

[root@web56 ~]# make && make install

[root@web56 ~]# yum -y php-fpm-5.4.16-42.el7.x86\_64.rpm

[root@web56 ~]# yum -y install mariadb-server

**修改配置文件**

[root@web56 ~]# vim /usr/local/nginx/conf/nginx.conf

location ~ \.php$ {

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

#fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

include fastcgi.conf;

}

**启动服务**

[root@web56 ~]# /usr/local/nginx/sbin/nginx

[root@web56 ~]# ss -natpul | grep :80

[root@web56 ~]# systemctl restart php-fpm

[root@web56 ~]# systemctl enable php-fpm

[root@web56 ~]# ss -natpul | grep nginx

[root@web56 ~]# ss -natpul | grep :9000

[root@web56 ~]# /usr/local/nginx/sbin/nginx -t

**//测试原配置文件的修改是否正确，正确输出如下：**



**建立测试文件、测试配置**

[root@web56 ~]# echo 123 > /usr/local/nginx/html/test.html

[root@web56 ~]# curl <http://localhost/test.html>

[root@web56 ~]# cat /usr/local/nginx/html/test.php

<?php

echo "hello boy!";

?>

[root@web56 ~]# curl http://localhost/test.php

### 配置PHP支持Redis

**运行Redis服务**

**装包并启动服务**

[root@web56 ~]# tar -xf redis-4.0.8.tar.gz

[root@web56 ~]# cd redis-4.0.8/

[root@web56 redis-4.0.8]# make && make install

[root@web56 redis-4.0.8]# ./utils/install\_server.sh

//初始化，使用默认，一路回车确定

[root@web56 redis-4.0.8]# ss -natpul | grep :6379

**查看php是否支持redis：**

**方法一：**

[root@web56 ~]# php -m | grep redis

**方法二：**

[root@web56 ~]# vim /usr/local/nginx/html/test2.php

<?php

phpinfo();

?>

[root@room9pc01 ~]# firefox http://192.168.4.56/test2.php

**安装php扩展**

**装包：**

[root@web56 ~]# yum -y install autoconf automake

[root@web56 ~]# yum -y install php-devel-5.4.16-42.el7.x86\_64.rpm

[root@web56 ~]# tar -xf php-redis-2.2.4.tar.gz

[root@web56 ~]# cd phpredis-2.2.4/

[root@web56 phpredis-2.2.4]# /usr/bin/phpize

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

[root@web56 phpredis-2.2.4]# make

[root@web56 phpredis-2.2.4]# make install //编译安装获取模块路径

Installing shared extensions: /usr/lib64/php/modules/ **//模块路径**

[root@web56 phpredis-2.2.4]# ls /usr/lib64/php/modules/

**改配置：**

[root@web56 ~]# vim /etc/php.ini

728 extension\_dir = "/usr/lib64/php/modules/"

729 ; On windows:

730 extension = "redis.so"

[root@web56 ~]# systemctl restart php-fpm

[root@web56 ~]# php -m | grep -i redis

测试配置

[root@web56 ~]# vim /usr/local/nginx/html/redis.php

<?php

$redis = new redis();

$redis->connect('127.0.0.1',6379);

$redis->set('asd','123456');

echo $redis->get('asd');

?>

[root@web56 ~]# redis-cli

127.0.0.1:6379> KEYS \* //没有数据

(empty list or set)

[root@web56 ~]# curl <http://localhost/redis.php> //访问本机的网站服务

123456

[root@web56 ~]# redis-cli //连接redis

127.0.0.1:6379> keys \* //查看所有变量，有输出

1) "asd"

127.0.0.1:6379> get asd

"123456"

扩展：

[root@mysql55 ~]# mysql -uroot -p123asd...A

mysql> create database gamedb;

mysql> create table gamedb.user(

-> name char(15),

-> password char(6)

-> );

mysql> grant select,insert on gamedb.\* to admin@"192.168.4.56"

identified by '123asd...A';

[root@web56 ~]# mysql -uadmin -p123asd...A -h192.168.4.55

mysql> insert into gamedb.user values ("bob","123456");

mysql> select \* from gamedb.user;

[root@web56 ~]# vim /usr/local/nginx/html/index.html

<html>

<form action="reg.php" method="post">

<h1> reg login user info :</h1>

<h3>LoginName: </h3><input type="text" name="name" size="20" maxlength="255"/>

<br />

<br />

<h3>LoginPass: </h3><input type="text" name="password" size="20" maxlength="255"/>

<br />

<br />

<h3><input type="submit" name="submit" value="SUBMIT" /></h3></p>

</form>

</html>

[root@web56 ~]# vim /usr/local/nginx/html/reg.php

<?php

$servername = "192.168.4.55";

$username = "admin";

$password = "123qqq...A";

$dbname = "gamedb";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("link fail: " . $conn->connect\_error);

}else{echo "link mysql Ok"; echo " ";}

$username=$\_POST['name'];

$userpassword=$\_POST['password'];

$sql="insert into user (name,password) values ('$username','$userpassword')";

if($conn->query($sql)){echo "insert data ok"; }

$conn->close();

?>

[root@room9pc01 ~]# firefox http://192.168.4.56