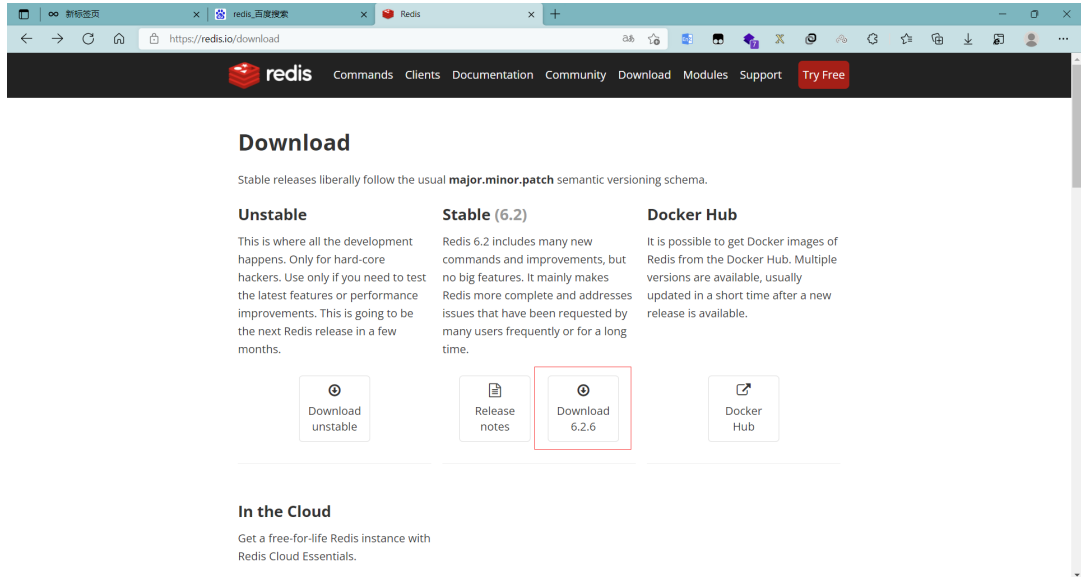


Redis 安装

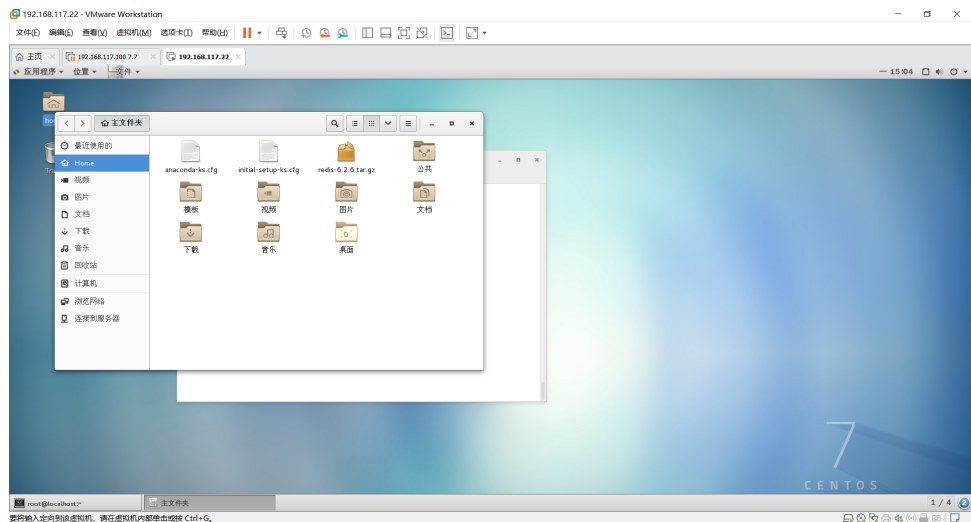
1. 下载软件包

找到redis官网，下载最新软件包[Redis官网下载](https://redis.io/download)



2. 安装redis 根据官方文档[重新开始快速 - 重新 \(redis.io\)](https://redis.io/docs/latest/en/quickstart/)

1. 将下载的Redis安装包拖入虚拟机中



2. 解压安装redis

```
1 [root@localhost ~]# tar -zxvf redis-6.2.6.tar.gz
```

```
root@localhost ~]# tar -zxvf redis-6.2.6.tar.gz
redis-6.2.6/
redis-6.2.6/.github/
redis-6.2.6/.github/ISSUE_TEMPLATE/
redis-6.2.6/.github/ISSUE_TEMPLATE/bug_report.md
redis-6.2.6/.github/ISSUE_TEMPLATE/crash_report.md
redis-6.2.6/.github/ISSUE_TEMPLATE/feature_request.md
redis-6.2.6/.github/ISSUE_TEMPLATE/other_stuff.md
redis-6.2.6/.github/ISSUE_TEMPLATE/question.md
redis-6.2.6/.github/workflows/
redis-6.2.6/.github/workflows/ci.yml
redis-6.2.6/.github/workflows/daily.yml
redis-6.2.6/.gitignore
redis-6.2.6/00-RELEASENOTES
redis-6.2.6/BUGS
redis-6.2.6/CONDUCT
redis-6.2.6/CONTRIBUTING
redis-6.2.6/COPYING
redis-6.2.6/INSTALL
redis-6.2.6/MANIFESTO
redis-6.2.6/Makefile
redis-6.2.6/README.md
redis-6.2.6/TLS.md
```

- 1 [root@localhost ~]# ls
- 2 anaconda-ks.cfg redis-6.2.6.tar.gz 模板 文档 桌面
- 3 initial-setup-ks.cfg redis-stable.tar.gz 视频 下载
- 4 redis-6.2.6 公共 图片 音乐
- 5 [root@localhost ~]# cd redis-6.2.6/
- 6 [root@localhost redis-6.2.6]# make

```
root@localhost:~/redis-6.2.6
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost ~]# ls
anaconda-ks.cfg redis-6.2.6.tar.gz 模板 文档 桌面
initial-setup-ks.cfg redis-stable.tar.gz 视频 下载
redis-6.2.6 公共 图片 音乐
[root@localhost ~]# cd redis-6.2.6/
[root@localhost redis-6.2.6]# make
```

3. 将命令创建软链接

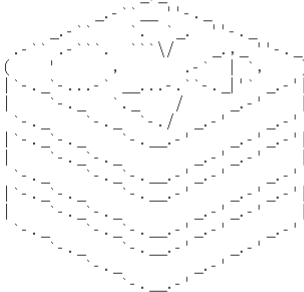
- 1 [root@localhost redis-6.2.6]# sudo cp src/redis-server /usr/local/bin/
- 2 [root@localhost redis-6.2.6]# sudo cp src/redis-cli /usr/local/bin/

4. 启动redis

- 1 root@localhost redis-6.2.6]# redis-server

```

[root@localhost redis-6.2.6]# redis-server
16051:C 01 Nov 2021 15:33:31.456 # o000o000o000o Redis is starting o000o000o000o
16051:C 01 Nov 2021 15:33:31.456 # Redis version=6.2.6, bits=64, commit=00000000, modified=0, pid=16051,
just started
16051:C 01 Nov 2021 15:33:31.456 # Warning: no config file specified, using the default config. In order
to specify a config file use redis-server /path/to/redis.conf
16051:M 01 Nov 2021 15:33:31.457 * Increased maximum number of open files to 10032 (it was originally se
t to 1024).
16051:M 01 Nov 2021 15:33:31.457 * monotonic clock: POSIX clock_gettime



Redis 6.2.6 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 16051

https://redis.io

16051:M 01 Nov 2021 15:33:31.458 # WARNING: The TCP backlog setting of 511 cannot be enforced because /p
roc/sys/net/core/somaxconn is set to the lower value of 128.
16051:M 01 Nov 2021 15:33:31.458 # Server initialized
16051:M 01 Nov 2021 15:33:31.458 # 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 reb
oot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
16051:M 01 Nov 2021 15:33:31.459 * Ready to accept connections

```

错误1

```

1 16164:M 01 Nov 2021 15:38:11.205 # WARNING: The TCP
backlog setting of 511 cannot be enforced because
/proc/sys/net/core/somaxconn is set to the lower value of
128.
2 16164:M 01 Nov 2021 15:38:11.205 # Server initialized
3 16164:M 01 Nov 2021 15:38:11.205 * Loading RDB produced by
version 6.2.6
4 16164:M 01 Nov 2021 15:38:11.205 * RDB age 164 seconds
5 16164:M 01 Nov 2021 15:38:11.205 * RDB memory usage when
created 0.77 Mb
6 16164:M 01 Nov 2021 15:38:11.205 # Done loading RDB, keys
loaded: 0, keys expired: 0.
7 16164:M 01 Nov 2021 15:38:11.205 * DB loaded from disk:
0.000 seconds
8 16164:M 01 Nov 2021 15:38:11.205 * Ready to accept
connections

```

```

1 16051:M 01 Nov 2021 15:33:31.458#警告：无法强制执行TCP积压设
置511，因为/proc/sys/net/core/somaxconn被设置为较低的值128。
2
3 16051:M 01 Nov 2021 15:33:31.458#服务器已初始化
4
5 16051:M 01 Nov 2021 15:33:31.458 35;警告超限#内存设置为0！在
内存不足的情况下，后台保存可能会失败。要解决此问题，请
将“vm.overmit_memory=1”添加到/etc/sysctl.conf，然后重新启动或
运行命令“sysctl vm.overmit_memory=1”，使其生效。
6
7 16051:M 01 Nov 2021 15:33:31.459*准备接受连接
8
9 ^C16051:信号处理程序（1635752127）收到SIGINT计划关闭。。。
10
11 16051:M 01 Nov 2021 15:35:27.944 35;用户请求关机。。。
12
13 16051:M 01 Nov 2021 15:35:27.944*退出前保存最终RDB快照。
14
15 16051:M 01 Nov 2021 15:35:27.945*DB保存在磁盘上
16

```

```
17 | 16051:M 01 Nov 2021 15:35:27.945#Redis现在准备退出，再见。。。

```

出现内存设置问题，根据redis报错提示

将参数“vm.overcommit_memory=1”添加到/etc/sysctl.conf，然后重新启动或运行命令“sysctl vm.overcommit_memory=1”，使其生效。

5. 将redis后台启动

```
1 | [root@localhost redis-6.2.6]# redis-server &

```

6. 测试redis连接

```
1 | [root@localhost redis-6.2.6]# redis-cli ping
2 | PONG

```

```
[root@localhost redis-6.2.6]# redis-cli ping
PONG
[root@localhost redis-6.2.6]# █

```

3. 配置redis的配置文件

1. 关闭redis

```
1 | [root@localhost redis-6.2.6]# redis-cli shutdown
2 | 16278:M 01 Nov 2021 15:49:39.438 # User requested shutdown...
3 | 16278:M 01 Nov 2021 15:49:39.438 * Saving the final RDB snapshot before exiting.
4 | 16278:M 01 Nov 2021 15:49:39.708 * DB saved on disk
5 | 16278:M 01 Nov 2021 15:49:39.708 # Redis is now ready to exit, bye bye...
6 | [1]+  完成                  redis-server

```

```
[root@localhost redis-6.2.6]# redis-cli shutdown
16278:M 01 Nov 2021 15:49:39.438 # User requested shutdown...
16278:M 01 Nov 2021 15:49:39.438 * Saving the final RDB snapshot before exiting.
16278:M 01 Nov 2021 15:49:39.708 * DB saved on disk
16278:M 01 Nov 2021 15:49:39.708 # Redis is now ready to exit, bye bye...
[1]+  完成                  redis-server
[root@localhost redis-6.2.6]# █

```

2. 创建存储 Redis 配置文件和数据的目录

```
1 | [root@localhost ~]# cd
2 | [root@localhost ~]# sudo mkdir /etc/redis
3 | [root@localhost ~]# sudo mkdir /var/redis

```

3. 将在utils目录下的Redis发行版中找到的init脚本复制到/etc/init.d中。我们建议使用运行此Redis实例的端口名来调用它

```
1 | [root@localhost ~]# cd redis-6.2.6/
2 | [root@localhost redis-6.2.6]# sudo cp utils/redis_init_script /etc/init.d/redis_6379

```

4. 编辑初始化脚本

```
1 | sudo vim /etc/init.d/redis_6379

```

```
#!/bin/sh
#
# Simple Redis init.d script conceived to work on Linux systems
# as it does use of the /proc filesystem.
### BEGIN INIT INFO
# Provides: redis_6379
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Redis data structure server
# Description: Redis data structure server. See https://redis.io
### END INIT INFO
REDISPORT=6379
EXEC=/usr/local/bin/redis-server
CLIEXEC=/usr/local/bin/redis-cli
PIDFILE=/var/run/redis_${REDISPORT}.pid
CONF="/etc/redis/${REDISPORT}.conf"
```

端口号默认为6379 不用修改

5. 使用端口号作为名称，将Redis发行版根目录中的模板配置文件复制到/etc/Redis/中

```
1 sudo cp redis.conf /etc/redis/6379.conf
```

6. 在/var/redis内创建一个目录，该目录将用作此redis实例的数据和工作目录

```
1 sudo mkdir /var/redis/6379
```

7. 编辑配置文件,确保执行以下更改:

- 将daemonize设置为yes（默认情况下设置为no）
- 将pidfile设置为（根据需要修改端口） /var/run/redis_6379.pid`
- 相应地更改端口。在我们的示例中，不需要它，因为默认端口已经是6379
- 设置您的首选日志级别。
- 将日志文件设置为 /var/log/redis_6379.log
- 将dir设置为/var/redis/6379（非常重要的一步！）

Edit the configuration file, making sure to perform the following changes:

- Set **daemonize** to yes (by default it is set to no).
- Set the **pidfile** to (modify the port if needed). /var/run/redis_6379.pid
- Change the **port** accordingly. In our example it is not needed as the default port is already 6379.
- Set your preferred **loglevel**.
- Set the **logfile** to /var/log/redis_6379.log
- Set the **dir** to /var/redis/6379 (very important step!)

1. 进入配置文件

```
1 [root@localhost redis-6.2.6]# vim /etc/redis/6379.conf
```

8. 修改配置文件

9. 使用搜索,搜索需要更改的配置项

```
# Redis configuration file example.
#
# Note that in order to read the configuration file, Redis must be
# started with the file path as first argument:
#
# ./redis-server /path/to/redis.conf
#
# Note on units: when memory size is needed, it is possible to specify
# it in the usual form of 1k 5GB 4M and so forth:
#
# 1k => 1000 bytes
# 1kb => 1024 bytes
# 1m => 1000000 bytes
# 1mb => 1024*1024 bytes
# 1g => 1000000000 bytes
# 1gb => 1024*1024*1024 bytes
#
# units are case insensitive so 1GB 1Gb 1gB are all the same.
##### INCLUDES #####
# Include one or more other config files here. This is useful if you
# have a standard template that goes to all Redis servers but also need
# to customize a few per-server settings. Include files can include
# other files, so use this wisely.
#
# Note that option "include" won't be rewritten by command "CONFIG REWRITE"
# from admin or Redis Sentinel. Since Redis always uses the last processed
# line as value of a configuration directive, you'd better put includes
# at the beginning of this file to avoid overwriting config change at runtime.
#
# If instead you are interested in using includes to override configuration
/daemonize

# By default Redis does not run as a daemon. Use 'yes' if you need it.
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.
# When Redis is supervised by upstart or systemd, this parameter has no impact.
daemonize yes
```

1 257 daemonize yes #257行的此配置项 更改为yes

10. 修改pidfile文件位置

1 289 pidfile /var/run/redis_6379.pid

11. 修改日志级别

1 297 loglevel notice

12. 修改日志文件位置

1 302 logfile "/var/log/redis_6379.log"

13. 修改目录位置

1 454 dir /var/redis/6379

14. 修改默认IP

1 75 bind 192.168.117.22

8. 使用以下命令将新的Redis init脚本添加到所有默认运行级别(设置开机自启动)

```
1 [root@localhost redis-6.2.6]# systemctl enable redis_6379
2 redis_6379.service is not a native service, redirecting to
  /sbin/chkconfig.
3 Executing /sbin/chkconfig redis_6379 on
```

9. 启动redis

```
1 [root@localhost redis-6.2.6]# sudo /etc/init.d/redis_6379
  start
2 Starting Redis server...
```

10. 关闭,开启,连接数据库

```
1 [root@localhost redis-6.2.6]# redis-cli -h 192.168.117.22
shutdown
2
3 [root@localhost redis-6.2.6]# sudo /etc/init.d/redis_6379
start
4 Starting Redis server...
5
6 [root@localhost redis-6.2.6]# redis-cli -h 192.168.117.22 -p
6379
7 192.168.117.22:6379>
```

11. 测试方法

确保一切按预期进行：

- 尝试使用redis cli ping您的实例
- 使用redis cli save执行测试保存，并检查转储文件是否正确存储到/var/redis/6379/（您应该找到一个名为dump.rdb的文件）。
- 检查您的Redis实例是否正确地登录到日志文件中。
- 如果这是一台新机器，您可以毫无问题地试用它，请确保在重新启动后，一切仍然正常。