tai34tw / **notes**

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master

notes / DB / Redis / RedisInstall / **redis-6.0.5-install-centOS7.md**

yt_tsai debug

0 contributors

Raw Blame

```
448 lines (406 sloc)   17.3 KB
```

Redis-6.0.5安裝-以CentOS 7為例

2020.06.23 蔡元泰製

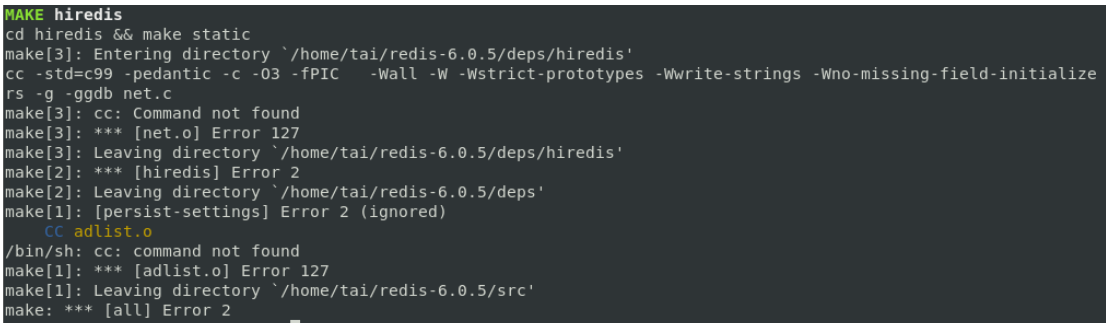
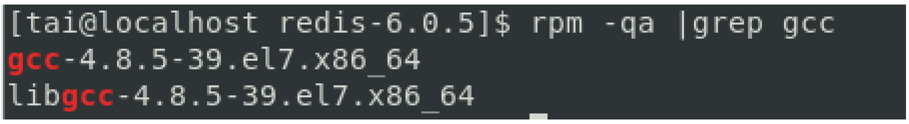
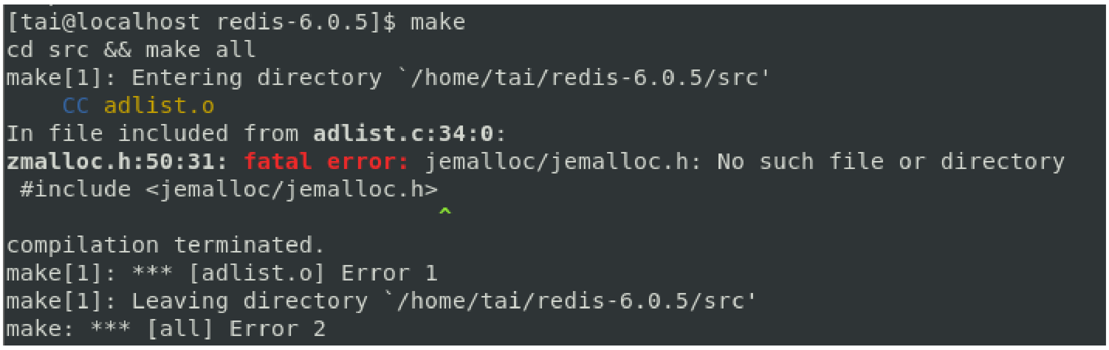
目錄

- 安裝
- 測試
- 啟動
- 使用內建客戶端與Redis溝通
- 快速執行
- 建議安裝補充
- 關閉Server
- 開啟外部訪問
- 管理工具(Windows 10為例)
- 參考來源

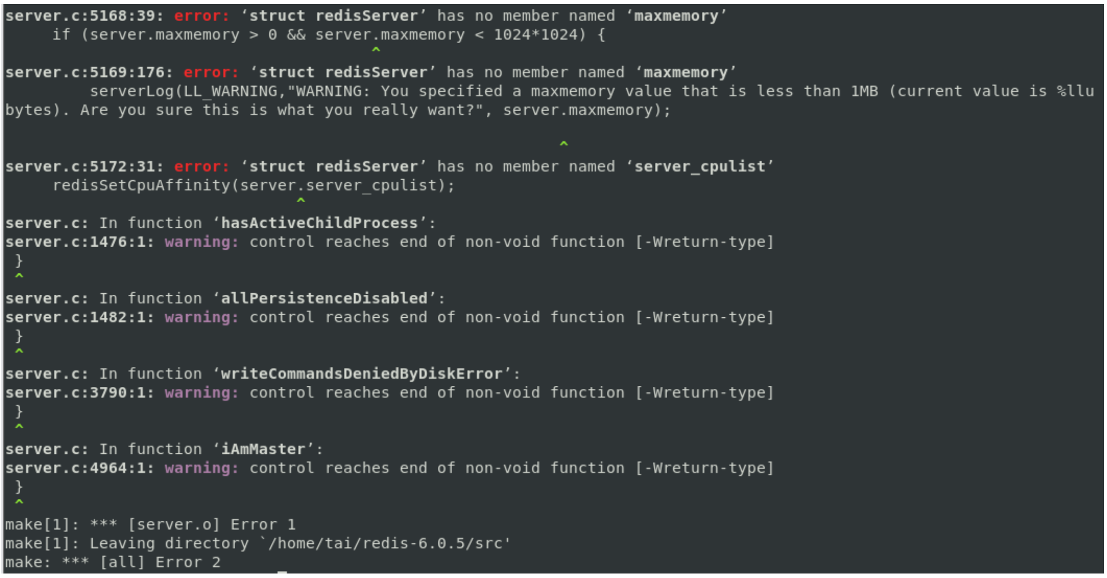
安裝

使用以下指令下載，提取和編譯Redis： [1] [2]

```
$ cd /opt
$ wget http://download.redis.io/releases/redis-6.0.5.tar.gz
$ tar xzf redis-6.0.5.tar.gz
$ cd redis-6.0.5
$ make distclean && make
```

> ## 可能錯誤:
>
> - cc: Command not found -> 未有GCC編譯器(C語言)，安裝即可(版本更新如後). [3] [4]
>
>   
>
>   ```
>   MAKE hiredis
>   cd hiredis && make static
>   make[3]: Entering directory `/home/tai/redis-6.0.5/deps/hiredis'
>   cc -std=c99 -pedantic -c -O3 -fPIC   -Wall -W -Wstrict-prototypes -Wwrite-strings -Wno-missing-field-initialize
>   rs -g -ggdb net.c
>   make[3]: cc: Command not found
>   make[3]: *** [net.o] Error 127
>   make[3]: Leaving directory `/home/tai/redis-6.0.5/deps/hiredis'
>   make[2]: *** [hiredis] Error 2
>   make[2]: Leaving directory `/home/tai/redis-6.0.5/deps'
>   make[1]: [persist-settings] Error 2 (ignored)
>       CC adlist.o
>   /bin/sh: cc: command not found
>   make[1]: *** [adlist.o] Error 127
>   make[1]: Leaving directory `/home/tai/redis-6.0.5/src'
>   make: *** [all] Error 2
>   ```
>
>   執行:
>
>   ```
>   $ cd redis-6.0.5
>   $ [sudo] yum install gcc
>   $ rpm -qa |grep gcc # 驗證gcc是否安裝成功
>   ```
>
>   
>
>   ```
>   [tai@localhost redis-6.0.5]$ rpm -qa |grep gcc
>   gcc-4.8.5-39.el7.x86_64
>   libgcc-4.8.5-39.el7.x86_64
>   ```
>
>   ```
>   $ make
>   ```
>
> - jemalloc/jemalloc.h: No such file or directory -> 上次編譯有殘留文件，需清理後再重新編譯. [5]
>
>   
>
>   ```
>   [tai@localhost redis-6.0.5]$ make
>   cd src && make all
>   make[1]: Entering directory `/home/tai/redis-6.0.5/src'
>       CC adlist.o
>   In file included from adlist.c:34:0:
>   zmalloc.h:50:31: fatal error: jemalloc/jemalloc.h: No such file or directory
>    #include <jemalloc/jemalloc.h>
>                                  ^
>   compilation terminated.
>   make[1]: *** [adlist.o] Error 1
>   make[1]: Leaving directory `/home/tai/redis-6.0.5/src'
>   make: *** [all] Error 2
>   ```

- server.c:5172:31: error: ‘struct redisServer’ has no member named 'XXXXX' -> gcc版本不夠新(CentOS 7 默認安裝4.8.5)，升級至gcc 9. [6]



```
server.c:5168:39: error: ‘struct redisServer’ has no member named ‘maxmemory’
     if (server.maxmemory > 0 && server.maxmemory < 1024*1024) {
                                       ^
server.c:5169:176: error: ‘struct redisServer’ has no member named ‘maxmemory’
         serverLog(LL_WARNING,"WARNING: You specified a maxmemory value that is less than 1MB (current value is %llu
bytes). Are you sure this is what you really want?", server.maxmemory);

                                                           ^
server.c:5172:31: error: ‘struct redisServer’ has no member named ‘server_cpulist’
     redisSetCpuAffinity(server.server_cpulist);
                               ^
server.c: In function ‘hasActiveChildProcess’:
server.c:1476:1: warning: control reaches end of non-void function [-Wreturn-type]
 }
 ^
server.c: In function ‘allPersistenceDisabled’:
server.c:1482:1: warning: control reaches end of non-void function [-Wreturn-type]
 }
 ^
server.c: In function ‘writeCommandsDeniedByDiskError’:
server.c:3790:1: warning: control reaches end of non-void function [-Wreturn-type]
 }
 ^
server.c: In function ‘iAmMaster’:
server.c:4964:1: warning: control reaches end of non-void function [-Wreturn-type]
 }
 ^
make[1]: *** [server.o] Error 1
make[1]: Leaving directory `/home/tai/redis-6.0.5/src'
make: *** [all] Error 2
```

執行:

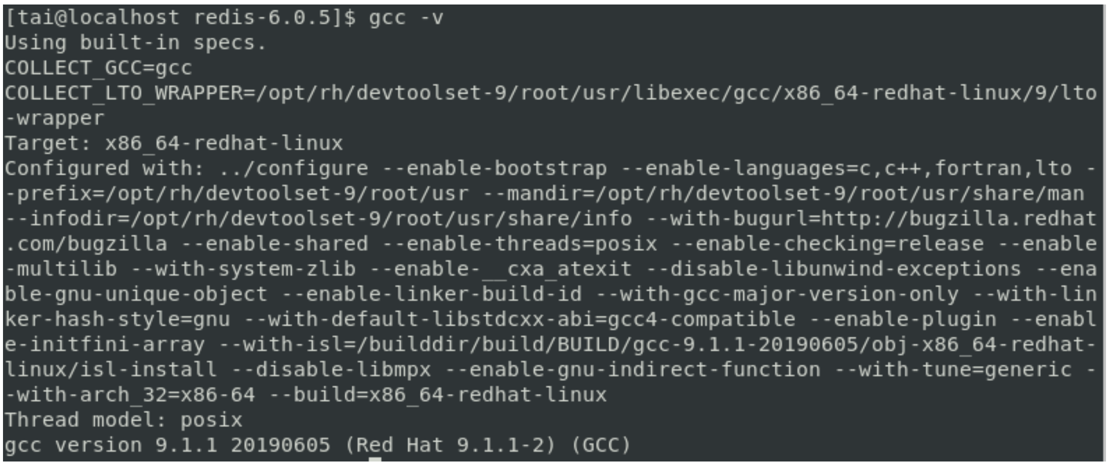
```
$ cd redis-6.0.5
$ make distclean # 清除編譯生成的文件.
$ sudo yum -y install centos-release-scl
$ sudo yum -y install devtoolset-9-gcc devtoolset-9-gcc-c++ devtoolset-9
$ scl enable devtoolset-9 bash # scl指令啟用只是臨時的，退出shell或重新打
```

執行以永久使用. [7]

```
$ sudo sh -c "echo source /opt/rh/devtoolset-9/enable >> /etc/profile"
```

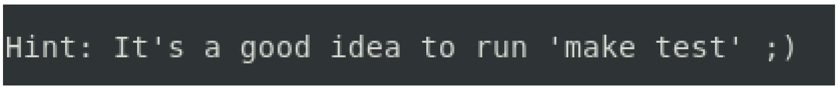
重打shell (或重開機)，再次編譯.

```
$ gcc -v  # 驗證gcc版本.
```



```
[tai@localhost redis-6.0.5]$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/opt/rh/devtoolset-9/root/usr/libexec/gcc/x86_64-redhat-linux/9/lto
-wrapper
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,lto -
-prefix=/opt/rh/devtoolset-9/root/usr --mandir=/opt/rh/devtoolset-9/root/usr/share/man
--infodir=/opt/rh/devtoolset-9/root/usr/share/info --with-bugurl=http://bugzilla.redhat
.com/bugzilla --enable-shared --enable-threads=posix --enable-checking=release --enable
-multilib --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --ena
ble-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --with-lin
ker-hash-style=gnu --with-default-libstdcxx-abi=gcc4-compatible --enable-plugin --enabl
e-initfini-array --with-isl=/builddir/build/BUILD/gcc-9.1.1-20190605/obj-x86_64-redhat-
linux/isl-install --disable-libmpx --enable-gnu-indirect-function --with-tune=generic -
-with-arch_32=x86-64 --build=x86_64-redhat-linux
Thread model: posix
gcc version 9.1.1 20190605 (Red Hat 9.1.1-2) (GCC)
```

安裝成功:

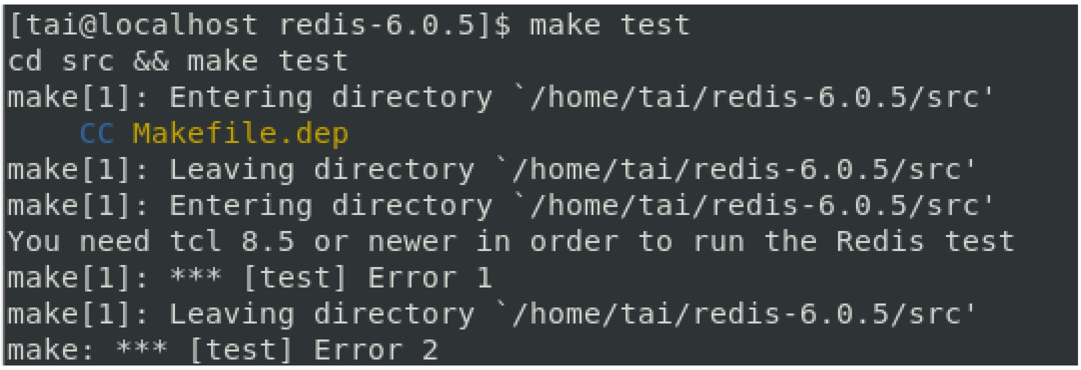
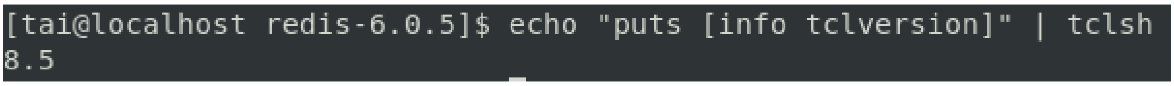
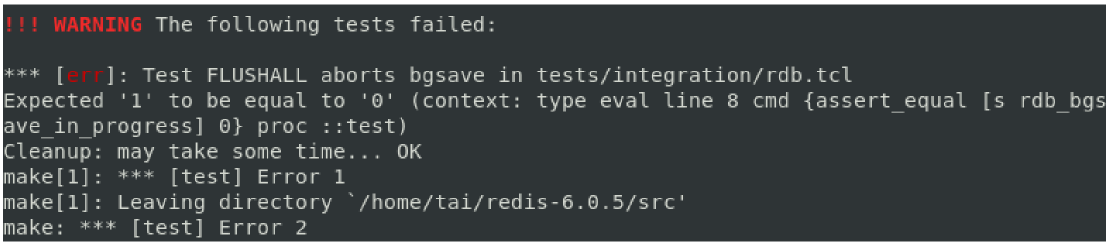
> 
>
> ```
> Hint: It's a good idea to run 'make test' ;)
> ```

回到目錄

測試

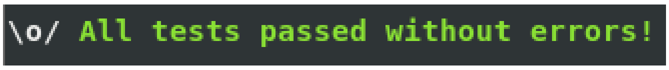
使用以下指令執行Redis運行測試：

```
$ cd redis-6.0.5
$ make test
```

> ### 可能錯誤:
>
> - You need tcl 8.5 or newer in order to run the Redis test -> 未安裝tcl 8.5. [8]
>
>   
>
>   ```
>   [tai@localhost redis-6.0.5]$ make test
>   cd src && make test
>   make[1]: Entering directory `/home/tai/redis-6.0.5/src'
>       CC Makefile.dep
>   make[1]: Leaving directory `/home/tai/redis-6.0.5/src'
>   make[1]: Entering directory `/home/tai/redis-6.0.5/src'
>   You need tcl 8.5 or newer in order to run the Redis test
>   make[1]: *** [test] Error 1
>   make[1]: Leaving directory `/home/tai/redis-6.0.5/src'
>   make: *** [test] Error 2
>   ```
>
>   執行:
>
>   ```
>   $ cd redis-6.0.5
>   $ [sudo] yum install tcl
>   $ echo "puts [info tclversion]" | tclsh # 驗證tcl版本
>   ```
>
> 
>
> ```
> [tai@localhost redis-6.0.5]$ echo "puts [info tclversion]" | tclsh
> 8.5
> ```
>
> - *** [err]: Test FLUSHALL aborts bgsave in tests/integration/rdb.tcl -> 效能配置不足，增加效能配置即可 (測試無法通過，不代表Redis無法運行).[9]
>
>   
>
>   ```
>   !!! WARNING The following tests failed:
>
>   *** [err]: Test FLUSHALL aborts bgsave in tests/integration/rdb.tcl
>   Expected '1' to be equal to '0' (context: type eval line 8 cmd {assert_equal [s rdb_bgsave_in_progress] 0} proc ::test)
>   Cleanup: may take some time... OK
>   make[1]: *** [test] Error 1
>   make[1]: Leaving directory `/home/tai/redis-6.0.5/src'
>   make: *** [test] Error 2
>   ```
>
>   執行: 增加(虛擬機)記憶體及處理器效能，以下供參:
>
>   - 記憶體: 4G.

> - 處理器: 實體2核心，虛擬2核心，共4核心.
> - 硬碟: 40G.

運行測試通過:



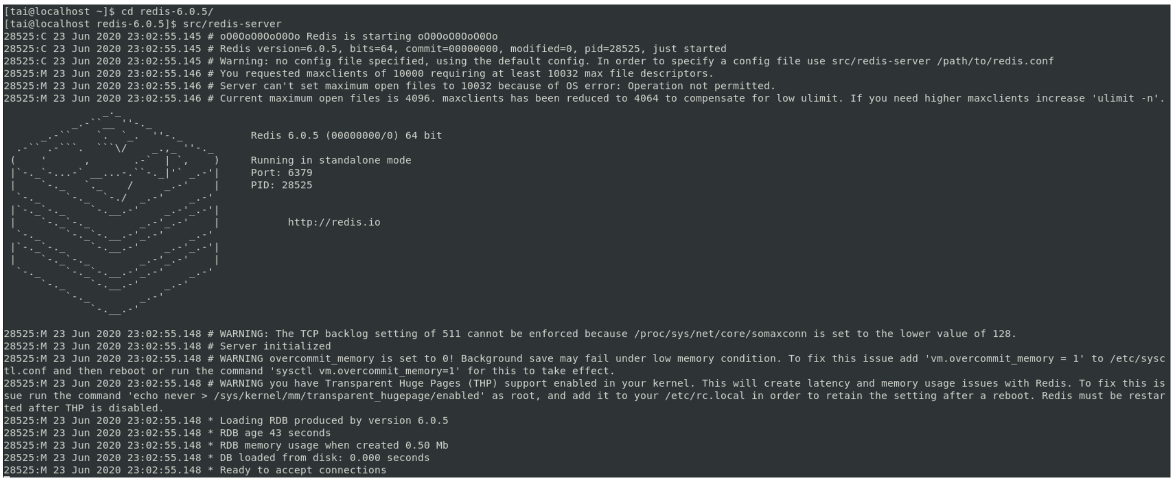
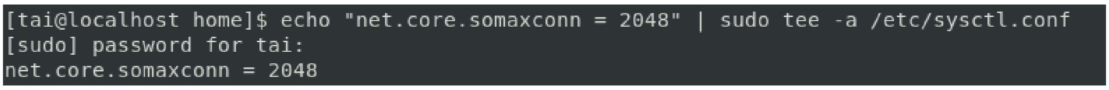
```
\o/ All tests passed without errors!
```

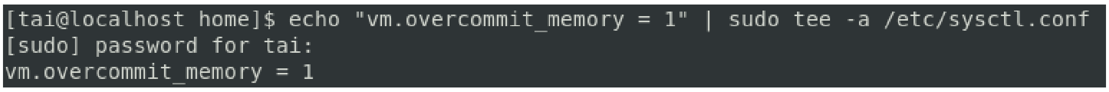
回到目錄

啟動

使用以下指令運行Redis：

```
$ cd redis-6.0.5
$ src/redis-server
```

> ## 可能錯誤 (但可能不影響運行):
>
> 
>
> ```
> [tai@localhost ~]$ cd redis-6.0.5/
> [tai@localhost redis-6.0.5]$ src/redis-server
> 28525:C 23 Jun 2020 23:02:55.145 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
> 28525:C 23 Jun 2020 23:02:55.145 # Redis version=6.0.5, bits=64, commit=00000000, modified=0, pid=28525, just started
> 28525:C 23 Jun 2020 23:02:55.145 # Warning: no config file specified, using the default config. In order to specify a config file use src/redis-server /path/to/redis.conf
> 28525:M 23 Jun 2020 23:02:55.146 # You requested maxclients of 10000 requiring at least 10032 max file descriptors.
> 28525:M 23 Jun 2020 23:02:55.146 # Server can't set maximum open files to 10032 because of OS error: Operation not permitted.
> 28525:M 23 Jun 2020 23:02:55.146 # Current maximum open files is 4096. maxclients has been reduced to 4064 to compensate for low ulimit. If you need higher maxclients increase 'ulimit -n'.
>
>                                 Redis 6.0.5 (00000000/0) 64 bit
>
>                                 Running in standalone mode
>                                 Port: 6379
>                                 PID: 28525
>
>                                      http://redis.io
>
> 28525:M 23 Jun 2020 23:02:55.148 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
> 28525:M 23 Jun 2020 23:02:55.148 # Server initialized
> 28525:M 23 Jun 2020 23:02:55.148 # 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/sysc
> tl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
> 28525:M 23 Jun 2020 23:02:55.148 # 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 is
> sue 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 restar
> ted after THP is disabled.
> 28525:M 23 Jun 2020 23:02:55.148 * Loading RDB produced by version 6.0.5
> 28525:M 23 Jun 2020 23:02:55.148 * RDB age 43 seconds
> 28525:M 23 Jun 2020 23:02:55.148 * RDB memory usage when created 0.50 Mb
> 28525:M 23 Jun 2020 23:02:55.148 * DB loaded from disk: 0.000 seconds
> 28525:M 23 Jun 2020 23:02:55.148 * Ready to accept connections
> ```
>
> - WARNING: The TCP backlog setting of 511 cannot be... (監聽佇列的長度預設128):
>   執行: [10]
>
>   ```
>   $ echo "net.core.somaxconn = 2048" | sudo tee -a /etc/sysctl.conf
>   ```
>
>   
>
>   ```
>   [tai@localhost home]$ echo "net.core.somaxconn = 2048" | sudo tee -a /etc/sysctl.conf
>   [sudo] password for tai:
>   net.core.somaxconn = 2048
>   ```
>
> - WARNING overcommit_memory is set to 0! (內存分配策略參數設置為0):
>   執行:
>
>   ```
>   $ echo "vm.overcommit_memory = 1" | sudo tee -a /etc/sysctl.conf
>   ```



```
[tai@localhost home]$ echo "vm.overcommit_memory = 1" | sudo tee -a /etc/sysctl.conf
[sudo] password for tai:
vm.overcommit_memory = 1
```

備註: 其他參數設定詳見: [11][12]

- WARNING you have Transparent Huge Pages (THP) support enabled in your kernel (你使用的是透明大頁，可能導致redis延遲和內存使用問題)....
  執行: [13]

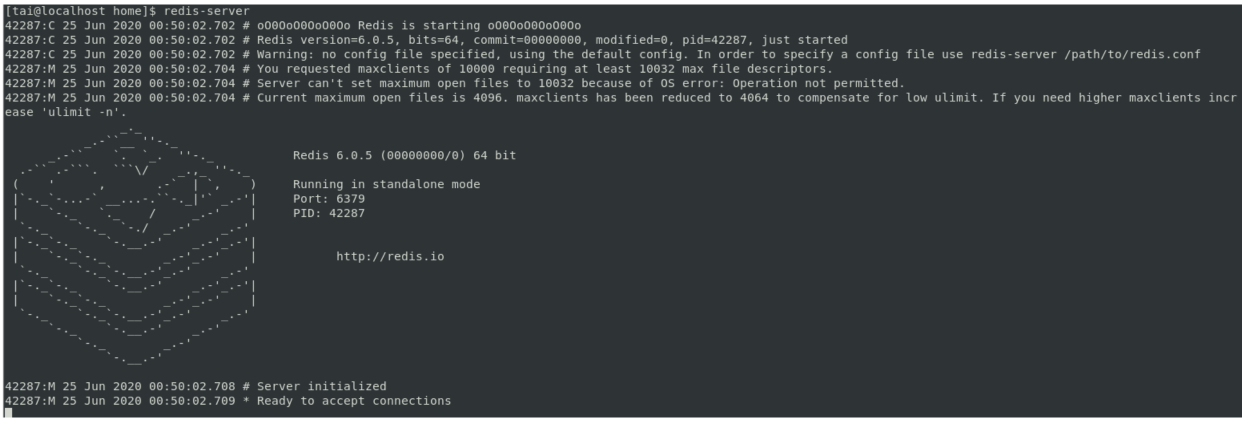
  - 暫時解決方法

    ```
    $ sudo su # 切換至root帳號，用sudo無法
    # echo never > /sys/kernel/mm/transparent_hugepage/enabled
    # exit # 切換回User帳號
    ```

  - 永久解決方法

    ```
    $ sudo su
    # vim /etc/rc.local
    # echo never > /sys/kernel/mm/transparent_hugepage/enabled
    # exit
    ```

運行成功:



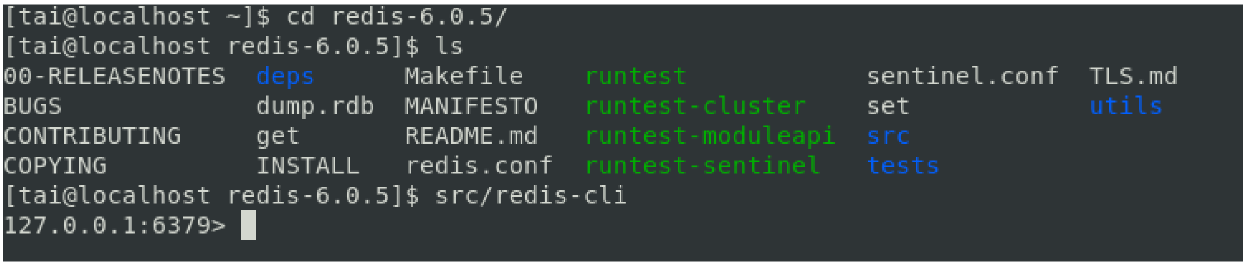
```
[tai@localhost home]$ redis-server
42287:C 25 Jun 2020 00:50:02.702 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
42287:C 25 Jun 2020 00:50:02.702 # Redis version=6.0.5, bits=64, commit=00000000, modified=0, pid=42287, just started
42287:C 25 Jun 2020 00:50:02.702 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
42287:M 25 Jun 2020 00:50:02.704 # You requested maxclients of 10000 requiring at least 10032 max file descriptors.
42287:M 25 Jun 2020 00:50:02.704 # Server can't set maximum open files to 10032 because of OS error: Operation not permitted.
42287:M 25 Jun 2020 00:50:02.704 # Current maximum open files is 4096. maxclients has been reduced to 4064 to compensate for low ulimit. If you need higher maxclients incr
ease 'ulimit -n'.
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 6.0.5 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 42287
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

42287:M 25 Jun 2020 00:50:02.708 # Server initialized
42287:M 25 Jun 2020 00:50:02.709 * Ready to accept connections
```

回到目錄

使用內建客戶端與Redis溝通

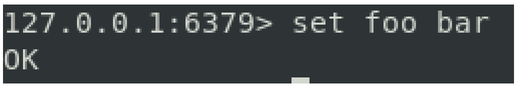
保持Redis運行，另外開啟shell，並使用下列指令與Redis溝通:

```
$ cd redis-6.0.5
$ src/redis-cli
```



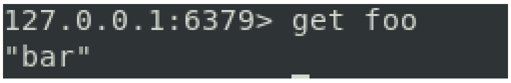
```
[tai@localhost ~]$ cd redis-6.0.5/
[tai@localhost redis-6.0.5]$ ls
00-RELEASENOTES  deps      Makefile    runtest             sentinel.conf  TLS.md
BUGS             dump.rdb  MANIFESTO   runtest-cluster     set            utils
CONTRIBUTING     get       README.md   runtest-moduleapi   src
COPYING          INSTALL   redis.conf  runtest-sentinel    tests
[tai@localhost redis-6.0.5]$ src/redis-cli
127.0.0.1:6379> 
```

- 新增資料: > set foo bar



```
127.0.0.1:6379> set foo bar
OK
```

- 搜尋資料: > get foo



```
127.0.0.1:6379> get foo
"bar"
```

回到目錄

快速執行

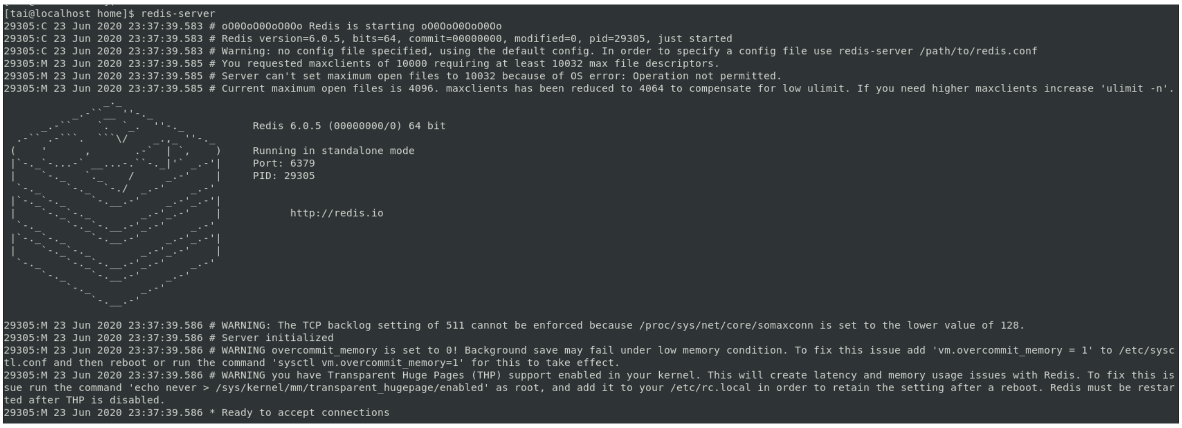
使用以下指令將主程式複製到/usr/local/bin/: [4]

```
$ cd redis-6.0.5/
$ sudo cp src/redis-server /usr/local/bin/
$ sudo cp src/redis-cli /usr/local/bin/
```

可以直接無視現在的目錄，直接執行redis的server與client 執行:

- server端:

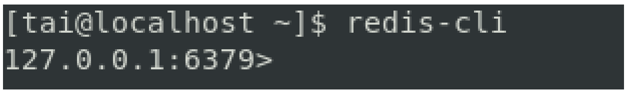
```
$ redis-server
```



```
[tai@localhost home]$ redis-server
29305:C 23 Jun 2020 23:37:39.583 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
29305:C 23 Jun 2020 23:37:39.583 # Redis version=6.0.5, bits=64, commit=00000000, modified=0, pid=29305, just started
29305:C 23 Jun 2020 23:37:39.583 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
29305:M 23 Jun 2020 23:37:39.585 # You requested maxclients of 10000 requiring at least 10032 max file descriptors.
29305:M 23 Jun 2020 23:37:39.585 # Server can't set maximum open files to 10032 because of OS error: Operation not permitted.
29305:M 23 Jun 2020 23:37:39.585 # Current maximum open files is 4096. maxclients has been reduced to 4064 to compensate for low ulimit. If you need higher maxclients increase 'ulimit -n'.

                                  Redis 6.0.5 (00000000/0) 64 bit

                                  Running in standalone mode
                                  Port: 6379
                                  PID: 29305

                                       http://redis.io

29305:M 23 Jun 2020 23:37:39.586 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
29305:M 23 Jun 2020 23:37:39.586 # Server initialized
29305:M 23 Jun 2020 23:37:39.586 # 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/sysc
tl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
29305:M 23 Jun 2020 23:37:39.586 # 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 is
sue 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 restar
ted after THP is disabled.
29305:M 23 Jun 2020 23:37:39.586 * Ready to accept connections
```

- 客戶端:

```
$ redis-cli
```



```
[tai@localhost ~]$ redis-cli
127.0.0.1:6379>
```

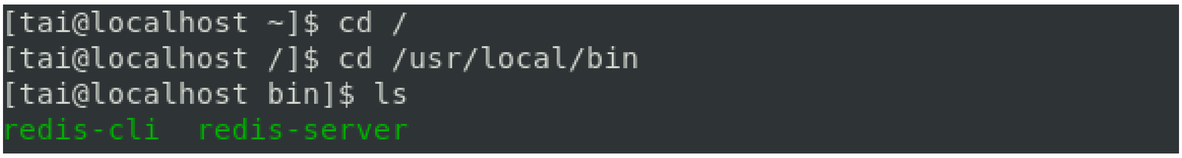
回到目錄

建議安裝補充

使用以下指令更適當地安裝Redis，以保存數據：[14]

1. 假設已經將redis-server和redis-cli可執行文件複製到/usr/local/bin下，可透過以下指令檢查.

```
$ cd /
$ cd /usr/local/bin
$ ls
```



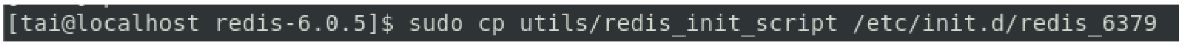
```
[tai@localhost ~]$ cd /
[tai@localhost /]$ cd /usr/local/bin
[tai@localhost bin]$ ls
redis-cli  redis-server
```

2. 創建一個目錄，用於存儲您Redis配置文件和數據：

```
$ cd /
$ sudo mkdir /etc/redis
$ sudo mkdir /var/redis
```

3. 找到當初下載Redis檔案處之初始化腳本(以「安裝」之redis-6.0.5為範例)，複製至/etc/init.d/下，官方建議以port為命名.

```
$ sudo cp utils/redis_init_script /etc/init.d/redis_6379
```

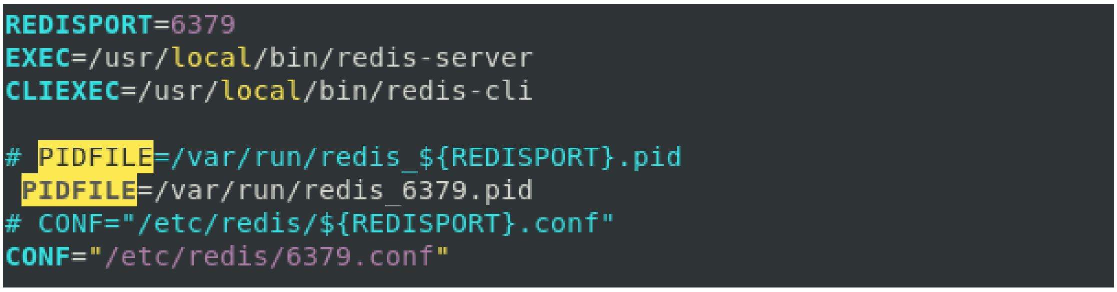


```
[tai@localhost redis-6.0.5]$ sudo cp utils/redis_init_script /etc/init.d/redis_6379
```

4. 編輯其腳本

```
$ sudo vim /etc/init.d/redis_6379
```

- 依據port修改REDISPORT, 其PIDFILE和CONF路徑皆取決於port (例如6379).

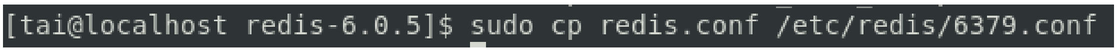


```
REDISPORT=6379
EXEC=/usr/local/bin/redis-server
CLIEXEC=/usr/local/bin/redis-cli

# PIDFILE=/var/run/redis_${REDISPORT}.pid
 PIDFILE=/var/run/redis_6379.pid
# CONF="/etc/redis/${REDISPORT}.conf"
CONF="/etc/redis/6379.conf"
```

5. 找到當初下載Redis檔案處(同步驟1)之配置文件模板, 複製到/etc/redis /中

   - 名稱與port相同(例如6379)

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



```
[tai@localhost redis-6.0.5]$ sudo cp redis.conf /etc/redis/6379.conf
```

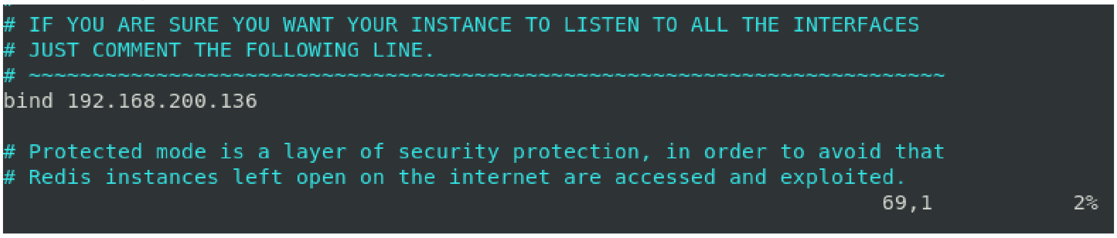
6. 創建一個目錄，供Redis之數據和工作目錄使用

```
$ sudo mkdir /var/redis/6379
```

7. 編輯配置文件，確保執行以下更改(直接google翻譯) (路人翻譯版, 請詳見: https://kknews.cc/zh-tw/code/y326ymk.html): [14][15]

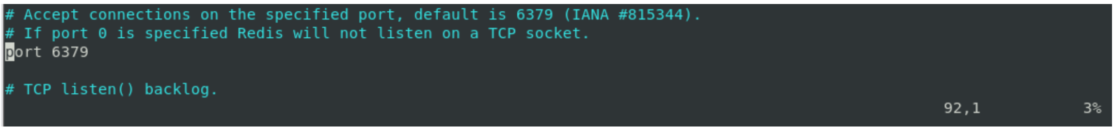
```
$ sudo vim /etc/redis/6379.conf
```

   - line: 69 (選用,供外部訪問)
     將ip (預設127.0.0.1)修改為本(虛擬)機ip.
     綁定多個ip, 可使用bind 127.0.0.1 192.168.200.136



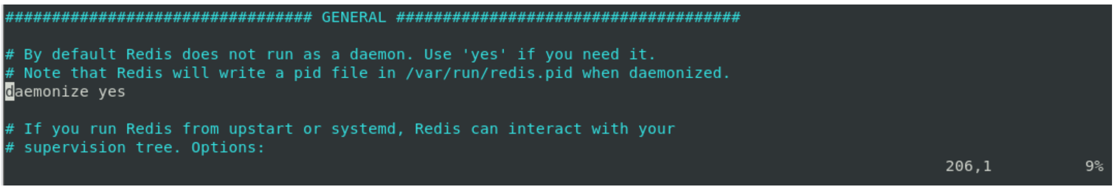
```
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
bind 192.168.200.136

# Protected mode is a layer of security protection, in order to avoid that
# Redis instances left open on the internet are accessed and exploited.
                                                     69,1          2%
```

   - line: 92 (更改port)
     在本件範例中，使用默認的port (6379)，因此不需要修改.



```
# Accept connections on the specified port, default is 6379 (IANA #815344).
# If port 0 is specified Redis will not listen on a TCP socket.
port 6379

# TCP listen() backlog.
                                                              92,1          3%
```

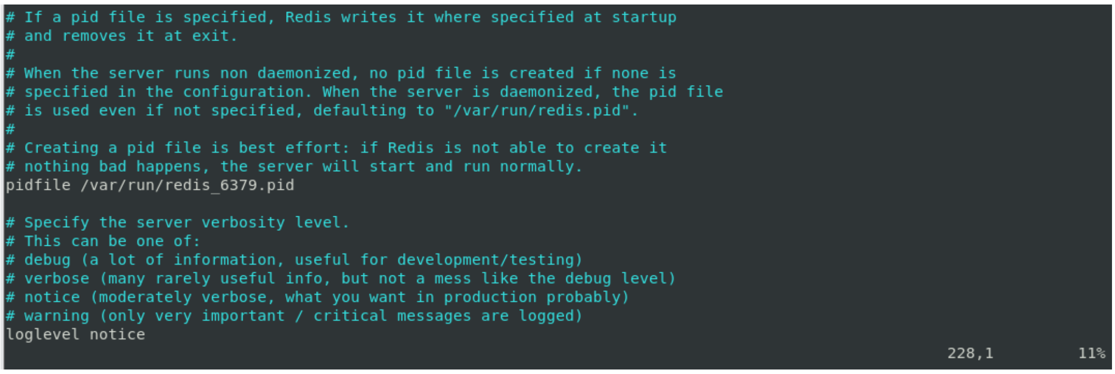
   - line: 206
     將守護程序(daemonize)設置為'yes' (默認情況下設置為'no').



```
################################# GENERAL #####################################

# 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.
daemonize yes

# If you run Redis from upstart or systemd, Redis can interact with your
# supervision tree. Options:
                                                              206,1          9%
```

- line: 228

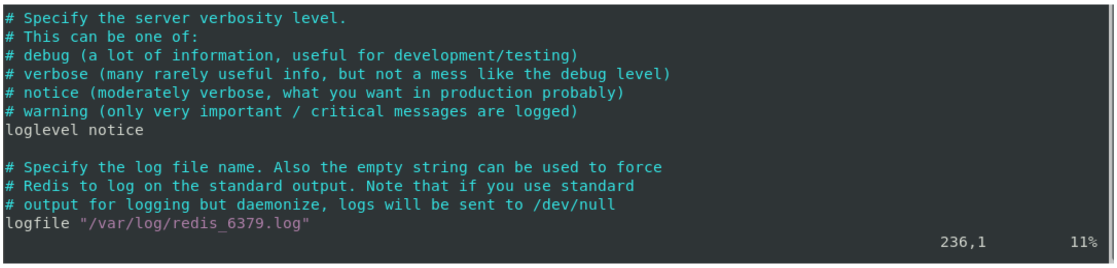
  將pidfile設置為/var/run/redis_6379.pid (如果需要，請修改port).



```
# If a pid file is specified, Redis writes it where specified at startup
# and removes it at exit.
#
# When the server runs non daemonized, no pid file is created if none is
# specified in the configuration. When the server is daemonized, the pid file
# is used even if not specified, defaulting to "/var/run/redis.pid".
#
# Creating a pid file is best effort: if Redis is not able to create it
# nothing bad happens, the server will start and run normally.
pidfile /var/run/redis_6379.pid

# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel notice
                                                              228,1         11%
```

- line: 236

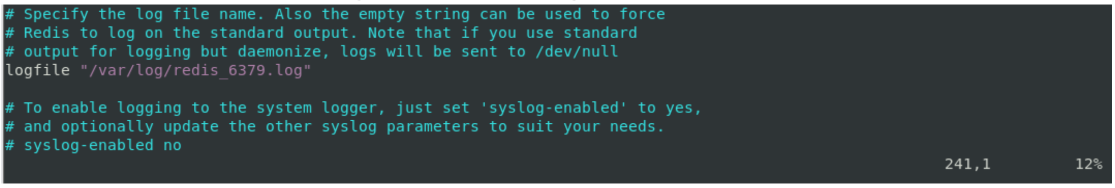
  設置日誌級別.



```
# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel notice

# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
# output for logging but daemonize, logs will be sent to /dev/null
logfile "/var/log/redis_6379.log"
                                                              236,1         11%
```

- line: 241 (日誌檔案的地址)

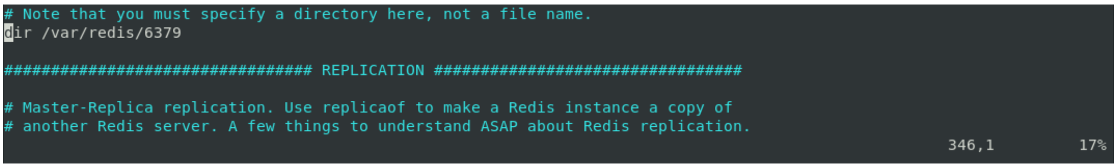
  將日誌文件設置為/var/log/redis_6379.log, 該路徑一定要是文件夾.



```
# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
# output for logging but daemonize, logs will be sent to /dev/null
logfile "/var/log/redis_6379.log"

# To enable logging to the system logger, just set 'syslog-enabled' to yes,
# and optionally update the other syslog parameters to suit your needs.
# syslog-enabled no
                                                              241,1         12%
```

- line: 346 (數據持久存放處)

  將目錄設置為/var/redis/6379 (非常重要的步驟！).



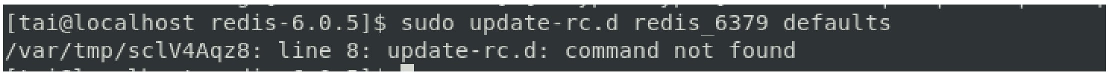
```
# Note that you must specify a directory here, not a file name.
dir /var/redis/6379

################################# REPLICATION #################################

# Master-Replica replication. Use replicaof to make a Redis instance a copy of
# another Redis server. A few things to understand ASAP about Redis replication.
                                                              346,1         17%
```

備註: 若不好找可用/搜尋關鍵字.

8. 將新的Redis腳本設為預設執行

```
$ sudo update-rc.d redis_6379 defaults
```

> ## 可能錯誤:



```
[tai@localhost redis-6.0.5]$ sudo update-rc.d redis_6379 defaults
/var/tmp/sclV4Aqz8: line 8: update-rc.d: command not found
```

> - sudo update-rc.d redis_6379 defaults 可能為Ubuntu 指令, CentOS不受用.
>   執行: [16]
> $ sudo chkconfig --add redis_6379
> $ sudo chkconfig redis_6379 on

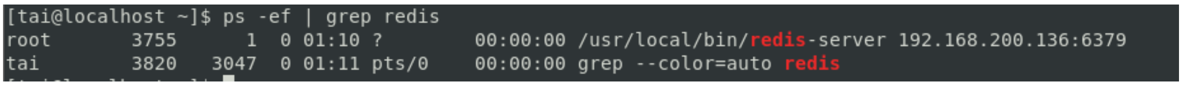
9. 執行修改後的Redis

```
$ sudo /etc/init.d/redis_6379 start
```

10. 檢視redis是否已經啟動

```
$ ps -ef | grep redis
```

- 啟動成功

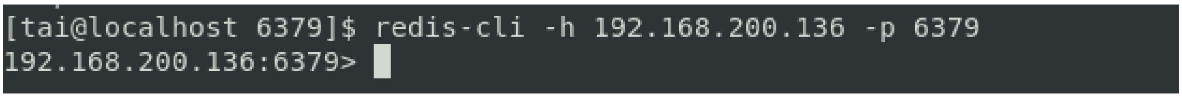


```
[tai@localhost ~]$ ps -ef | grep redis
root       3755      1  0 01:10 ?        00:00:00 /usr/local/bin/redis-server 192.168.200.136:6379
tai        3820   3047  0 01:11 pts/0    00:00:00 grep --color=auto redis
```

11. 透過客戶端檢視Redis是否執行成功

```
$ redis-cli -h 192.168.200.136 -p 6379
```

- 啟動成功



```
[tai@localhost 6379]$ redis-cli -h 192.168.200.136 -p 6379
192.168.200.136:6379>
```

回到目錄

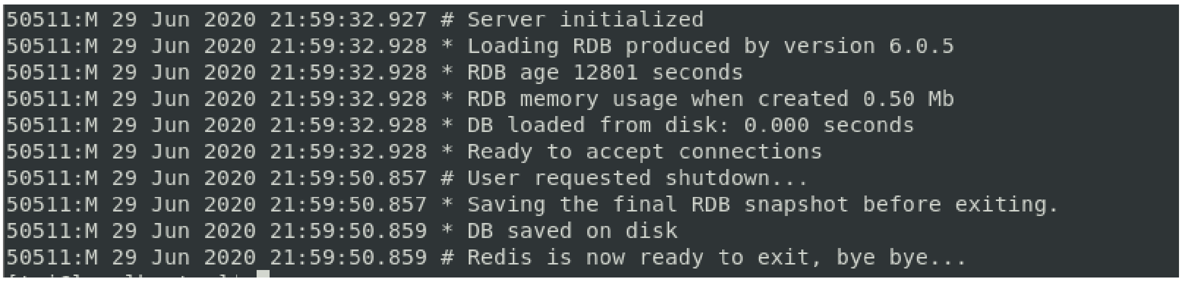
關閉Server

在客戶端使用以下指令關閉Server:

- 官方建議方法，確保在退出之前將數據保存:

```
$ redis-cli SHUTDOWN
```

查看原先開啟的Server端視窗

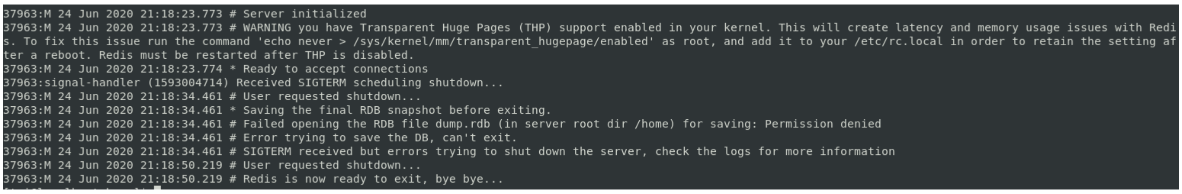


```
50511:M 29 Jun 2020 21:59:32.927 # Server initialized
50511:M 29 Jun 2020 21:59:32.928 * Loading RDB produced by version 6.0.5
50511:M 29 Jun 2020 21:59:32.928 * RDB age 12801 seconds
50511:M 29 Jun 2020 21:59:32.928 * RDB memory usage when created 0.50 Mb
50511:M 29 Jun 2020 21:59:32.928 * DB loaded from disk: 0.000 seconds
50511:M 29 Jun 2020 21:59:32.928 * Ready to accept connections
50511:M 29 Jun 2020 21:59:50.857 # User requested shutdown...
50511:M 29 Jun 2020 21:59:50.857 * Saving the final RDB snapshot before exiting.
50511:M 29 Jun 2020 21:59:50.859 * DB saved on disk
50511:M 29 Jun 2020 21:59:50.859 # Redis is now ready to exit, bye bye...
```

- 應該與上者類似，但未將數據保存(無法退出時，可用此指令): [17]

```
$ redis-cli SHUTDOWN NOSAVE
```

查看原先開啟的Server端視窗

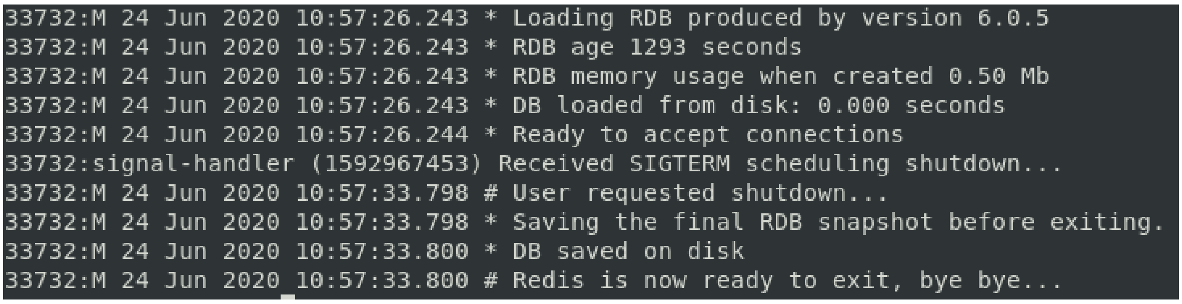


```
37963:M 24 Jun 2020 21:18:23.773 # Server initialized
37963:M 24 Jun 2020 21:18:23.773 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redi
s. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting af
ter a reboot. Redis must be restarted after THP is disabled.
37963:M 24 Jun 2020 21:18:23.774 * Ready to accept connections
37963:signal-handler (1593004714) Received SIGTERM scheduling shutdown...
37963:M 24 Jun 2020 21:18:34.461 # User requested shutdown...
37963:M 24 Jun 2020 21:18:34.461 * Saving the final RDB snapshot before exiting.
37963:M 24 Jun 2020 21:18:34.461 # Failed opening the RDB file dump.rdb (in server root dir /home) for saving: Permission denied
37963:M 24 Jun 2020 21:18:34.461 # Error trying to save the DB, can't exit.
37963:M 24 Jun 2020 21:18:34.461 # SIGTERM received but errors trying to shut down the server, check the logs for more information
37963:M 24 Jun 2020 21:18:50.219 # User requested shutdown...
37963:M 24 Jun 2020 21:18:50.219 # Redis is now ready to exit, bye bye...
```

- 正常退出，但有時候不成功: [18]

```
$ pkill redis-server
```

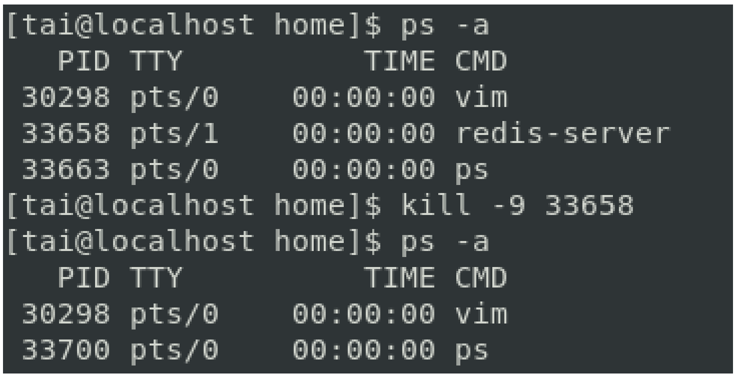
查看原先開啟的Server端視窗



```
33732:M 24 Jun 2020 10:57:26.243 * Loading RDB produced by version 6.0.5
33732:M 24 Jun 2020 10:57:26.243 * RDB age 1293 seconds
33732:M 24 Jun 2020 10:57:26.243 * RDB memory usage when created 0.50 Mb
33732:M 24 Jun 2020 10:57:26.243 * DB loaded from disk: 0.000 seconds
33732:M 24 Jun 2020 10:57:26.244 * Ready to accept connections
33732:signal-handler (1592967453) Received SIGTERM scheduling shutdown...
33732:M 24 Jun 2020 10:57:33.798 # User requested shutdown...
33732:M 24 Jun 2020 10:57:33.798 * Saving the final RDB snapshot before exiting.
33732:M 24 Jun 2020 10:57:33.800 * DB saved on disk
33732:M 24 Jun 2020 10:57:33.800 # Redis is now ready to exit, bye bye...
```

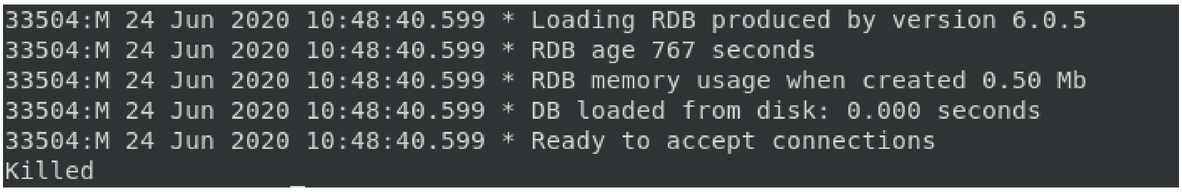
- 強制關閉(沒有log釋出): [19]

```
$ ps -a # 查看現有redis-server之PID
$ kill -9 3365 # 3365為該server之PID
```



```
[tai@localhost home]$ ps -a
   PID TTY          TIME CMD
 30298 pts/0    00:00:00 vim
 33658 pts/1    00:00:00 redis-server
 33663 pts/0    00:00:00 ps
[tai@localhost home]$ kill -9 33658
[tai@localhost home]$ ps -a
   PID TTY          TIME CMD
 30298 pts/0    00:00:00 vim
 33700 pts/0    00:00:00 ps
```

查看原先開啟的Server端視窗



```
33504:M 24 Jun 2020 10:48:40.599 * Loading RDB produced by version 6.0.5
33504:M 24 Jun 2020 10:48:40.599 * RDB age 767 seconds
33504:M 24 Jun 2020 10:48:40.599 * RDB memory usage when created 0.50 Mb
33504:M 24 Jun 2020 10:48:40.599 * DB loaded from disk: 0.000 seconds
33504:M 24 Jun 2020 10:48:40.599 * Ready to accept connections
Killed
```

回到目錄

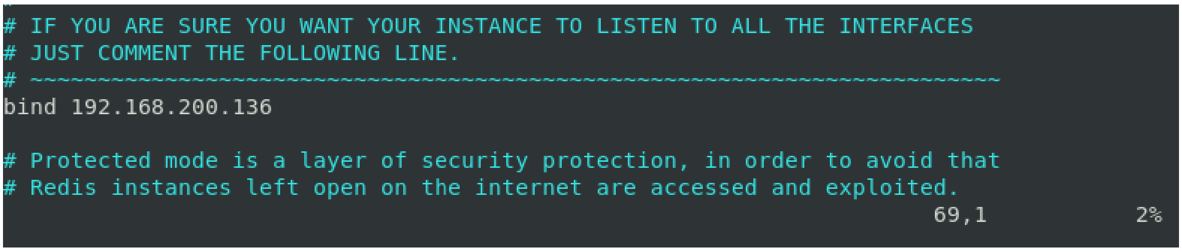
開啟外部訪問

使用以下指令使其Redis允許外部訪問:

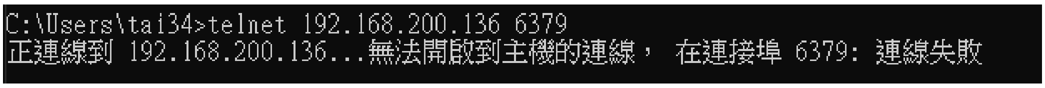
1. 編輯配置文件，確保執行以下更改(步驟同「建議安裝補充」(供外部訪問)： [15]

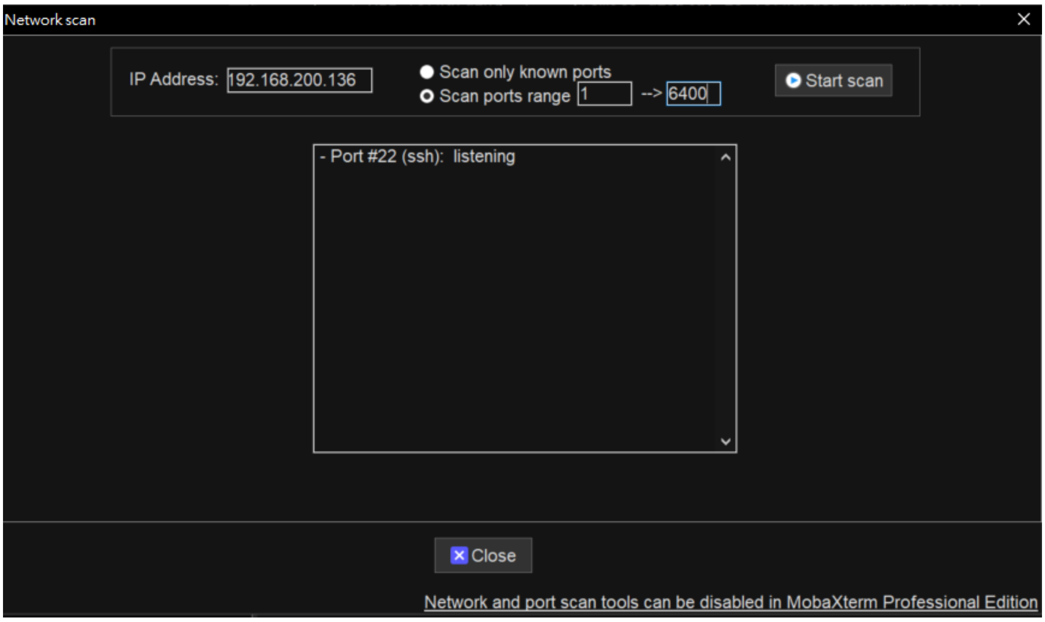
```
$ sudo vim /etc/redis/6379.conf
```

將ip (預設127.0.0.1)修改為本(虛擬)機ip.



```
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
bind 192.168.200.136

# Protected mode is a layer of security protection, in order to avoid that
# Redis instances left open on the internet are accessed and exploited.
                                                              69,1          2%
```

> ### 可能錯誤:
>
> - 外部無法連線 (以win 10為例)
>
>   ```
>   > telnet 192.168.200.136 6379
>   ```
>
>   
>
>   ```
>   C:\Users\tai34>telnet 192.168.200.136 6379
>   正連線到 192.168.200.136...無法開啟到主機的連線， 在連接埠 6379: 連線失敗
>   ```
>
>   透過MobaXterm之掃port工具(上方工具列Tools > Ports scanner), Cent OS 未外開Redis的port.


Network scan
✕
Start scan
Close
Network and port scan tools can be disabled in MobaXterm Professional Edition

IP Address: 192.168.200.136

○ Scan only known ports

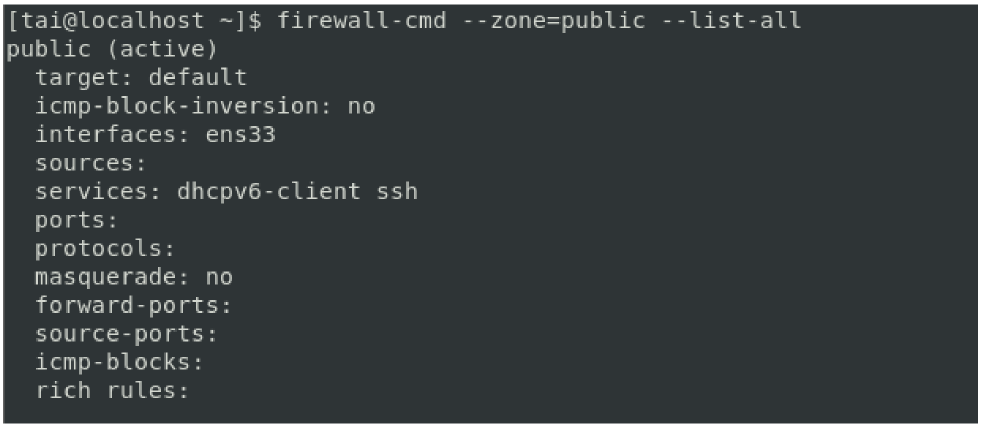
● Scan ports range 1 --> 6400

```
- Port #22 (ssh):  listening
```

執行:

a. 確認防火牆是否開啟

```
$ firewall-cmd --zone=public --list-all
```

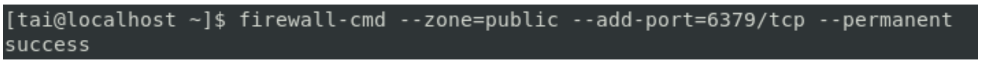


```
[tai@localhost ~]$ firewall-cmd --zone=public --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

b. 對外開放 6379 port

```
$ firewall-cmd --zone=public --add-port=6379/tcp --permanent
```

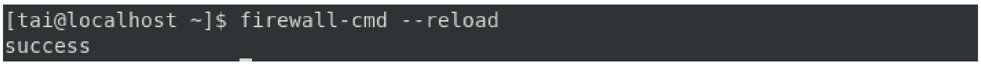
備註: --permanent 指定為永久設定，否則在 firewalld 重啟或是重新讀取設定，就會失效.



```
[tai@localhost ~]$ firewall-cmd --zone=public --add-port=6379/tcp --permanent
success
```

c. 重新讀取 firewall 設定

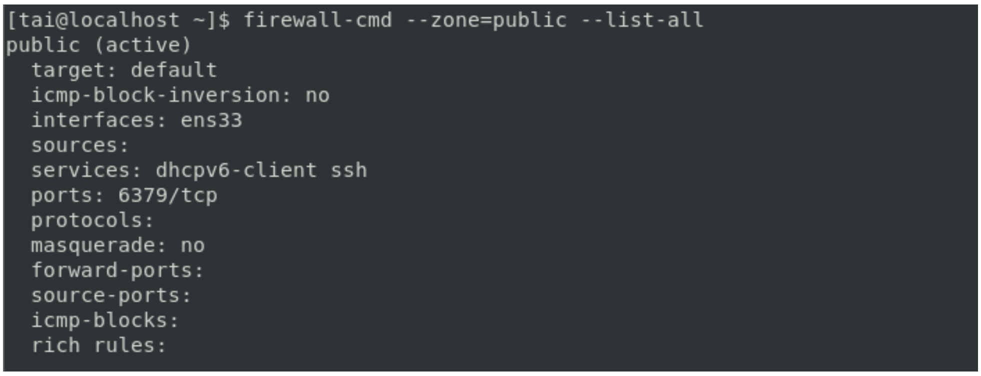
```
$ firewall-cmd --reload
```



```
[tai@localhost ~]$ firewall-cmd --reload
success
```

d. 與步驟1同，再檢查一次指定port是否在開放清單內

```
$ firewall-cmd --zone=public --list-all
```



```
[tai@localhost ~]$ firewall-cmd --zone=public --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports: 6379/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

e. 重新從本地端cmd測試

```
> telnet 192.168.200.136 6379
```

成功, 就是黑黑的一片.

回到目錄

管理工具(Windows 10為例)

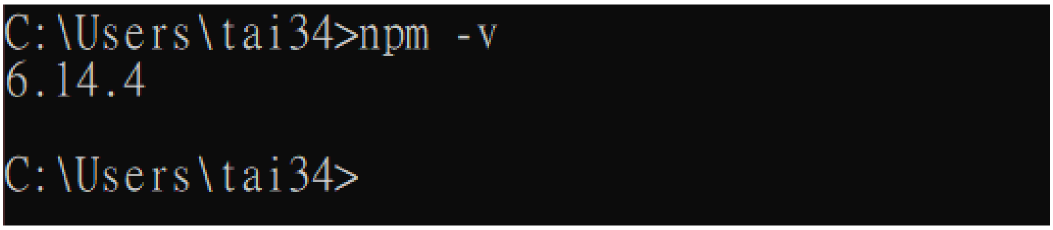
使用以下指令於cmd下操作:

- Redis-commander

  i. 檢查有無npm環境:

  ```
  > npm -v
  ```

  - 有

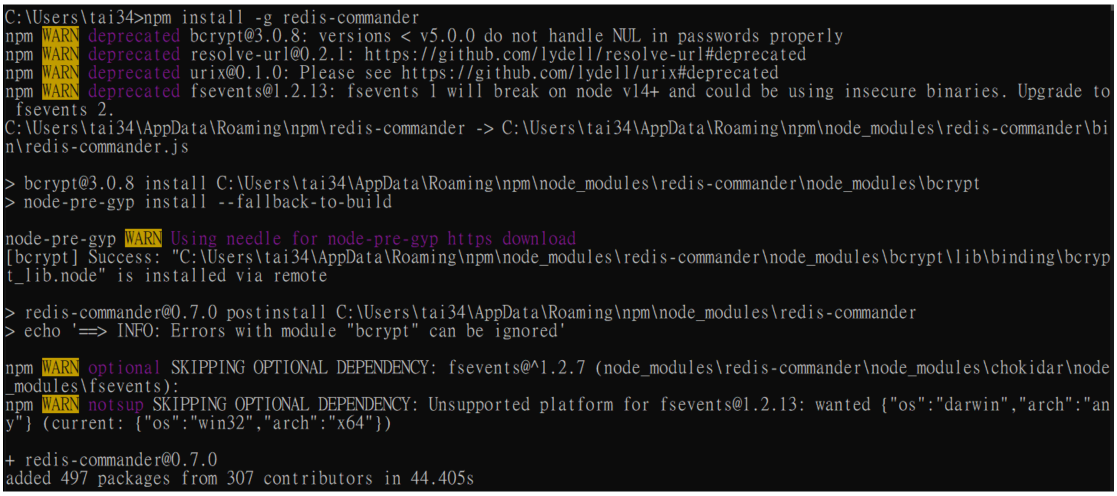
    ```
    C:\Users\tai34>npm -v
    6.14.4

    C:\Users\tai34>
    ```

  - 無

    請自行安裝.

  ii. 安裝redis-commander [19]

  ```
  > npm install -g redis-commander
  ```

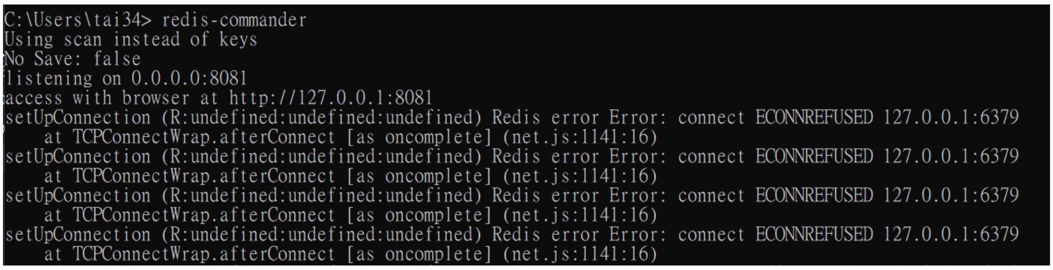


```
C:\Users\tai34>npm install -g redis-commander
npm WARN deprecated bcrypt@3.0.8: versions < v5.0.0 do not handle NUL in passwords properly
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated fsevents@1.2.13: fsevents 1 will break on node v14+ and could be using insecure binaries. Upgrade to
 fsevents 2.
C:\Users\tai34\AppData\Roaming\npm\redis-commander -> C:\Users\tai34\AppData\Roaming\npm\node_modules\redis-commander\bi
n\redis-commander.js

> bcrypt@3.0.8 install C:\Users\tai34\AppData\Roaming\npm\node_modules\redis-commander\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build

node-pre-gyp WARN Using needle for node-pre-gyp https download
[bcrypt] Success: "C:\Users\tai34\AppData\Roaming\npm\node_modules\redis-commander\node_modules\bcrypt\lib\binding\bcryp
t_lib.node" is installed via remote

> redis-commander@0.7.0 postinstall C:\Users\tai34\AppData\Roaming\npm\node_modules\redis-commander
> echo '==> INFO: Errors with module "bcrypt" can be ignored'

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.2.7 (node_modules\redis-commander\node_modules\chokidar\node
_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"an
y"} (current: {"os":"win32","arch":"x64"})

+ redis-commander@0.7.0
added 497 packages from 307 contributors in 44.405s
```

iii. 啟動redis-commander

- 連結本地端Server

```
> redis-commander
```

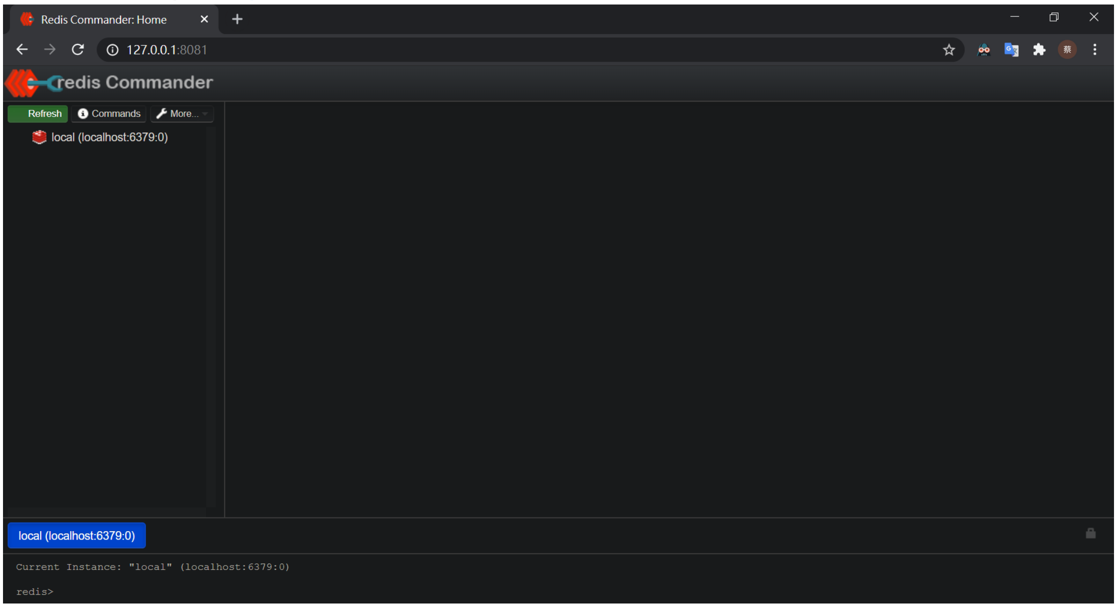


```
C:\Users\tai34> redis-commander
Using scan instead of keys
No Save: false
listening on 0.0.0.0:8081
access with browser at http://127.0.0.1:8081
setUpConnection (R:undefined:undefined:undefined) Redis error Error: connect ECONNREFUSED 127.0.0.1:6379
    at TCPConnectWrap.afterConnect [as oncomplete] (net.js:1141:16)
setUpConnection (R:undefined:undefined:undefined) Redis error Error: connect ECONNREFUSED 127.0.0.1:6379
    at TCPConnectWrap.afterConnect [as oncomplete] (net.js:1141:16)
setUpConnection (R:undefined:undefined:undefined) Redis error Error: connect ECONNREFUSED 127.0.0.1:6379
    at TCPConnectWrap.afterConnect [as oncomplete] (net.js:1141:16)
setUpConnection (R:undefined:undefined:undefined) Redis error Error: connect ECONNREFUSED 127.0.0.1:6379
    at TCPConnectWrap.afterConnect [as oncomplete] (net.js:1141:16)
```

- 連結至外部Redis

```
redis-commander --redis-host 192.168.200.136 --redis-port 6379
```

iv. 將上述畫面ip (http://127.0.0.1:8081)以瀏覽器開啟



```
Current Instance: "local" (localhost:6379:0)
redis>
```

備註:其他管理工具GUI: [20]

回到目錄

參考來源

1. https://redis.io/download
2. https://linuxize.com/post/how-to-install-and-configure-redis-on-centos-7/
3. https://www.itread01.com/content/1547765307.html
4. https://dotblogs.com.tw/jakeuj/2015/12/24/Redis
5. https://www.cnblogs.com/operationhome/p/10342258.html
6. https://www.cnblogs.com/sanduzxcvbnm/p/12955145.html
7. https://stackoverflow.com/questions/51030702/how-to-write-in-etc-profile-using-bash-permission-denied
8. https://www.cnblogs.com/zhaoshunjie/p/5907029.html
9. https://github.com/antirez/redis/issues/2126
10. https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/424386
11. https://access.redhat.com/documentation/zh-tw/red_hat_enterprise_linux/6/html/performance_tuning_guide/s-memory-captun
12. https://blog.csdn.net/hjx_1000/article/details/46412557
13. https://www.jianshu.com/p/7ca4b74c92be
14. https://redis.io/topics/quickstart
15. https://www.itread01.com/content/1546782362.html
16. https://unix.stackexchange.com/questions/65398/converting-update-rc-d-redis-6379-defaults-to-chkconfig-command
17. https://redis.io/commands/shutdown
18. http://www.jeepxie.net/article/964280.html
19. https://www.npmjs.com/package/redis-commander
20. https://redislabs.com/blog/so-youre-looking-for-the-redis-gui/

回到目錄