

## Instalación y configuración del Sistema Gestor de Bases de Datos NoSQL

La instalación y configuración de Redis se hizo sobre Ubuntu 20.04. Para ello se configuró WSL 2 (Windows Subsystem Linux) y se instaló la versión mencionada, también se lo hizo sobre una instalación nativa de Ubuntu.

Hay dos formas de instalar Redis:

- Directamente
- Mediante la descarga y compilación de los Binarios

En la primera, se ejecutan los siguientes pasos:

1. Se abre una terminal
2. Se adiciona el repositorios al índice apt:  
*sudo add-apt-repository ppa:redislabs/redis*
3. Se actualiza mediante  
*sudo apt-get update*
4. Se instala el paquete mediante  
*sudo apt-get install redis*
5. Para lanzar el servidor ejecutamos  
*redis-server*
6. Para ejecutar el cliente en otra terminal ejecutamos  
*redis-cli*

Para la segunda forma ejecutamos los siguientes pasos:

Se abre una terminal

1. Se abre una terminal
2. Para la descarga de los binarios se ejecuta el comando:  
*wget <https://download.redis.io/releases/redis-6.2.2.tar.gz>*
3. Luego se descomprime y desempaqueta los binarios con:

```
tar xzf redis-6.2.2.tar.gz
```

4. Nos ubicamos en el directorio donde se descomprimió los binarios:

```
cd redis-6.2.2
```

5. Luego compilamos con el comando para compilar

```
make
```

6. Los compilados se almacena en el directorio /src, por lo tanto para lanzar Redis se ejecuta

```
src/redis-server
```



```
ubuntu@INNO50FT2:~$ src/redis-server
1169:C 09 May 2021 14:53:38.635 # o000o000o000o Redis is starting o000o000o000o
1169:C 09 May 2021 14:53:38.635 # Redis version=6.2.3, bits=64, commit=00000000, modified=0, pid=1169, just started
1169:C 09 May 2021 14:53:38.635 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
1169:M 09 May 2021 14:53:38.636 # You requested maxclients of 10000 requiring at least 10032 max file descriptors.
1169:M 09 May 2021 14:53:38.636 # Server can't set maximum open files to 10032 because of OS error: Operation not permitted.
1169:M 09 May 2021 14:53:38.636 # Current maximum open files is 4096, maxclients has been reduced to 4064 to compensate for low ulimit. If you need higher maxclients in
crease 'ulimit -n'.
1169:M 09 May 2021 14:53:38.636 * monotonic clock: POSIX clock_gettime

Redis 6.2.3 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 1169
https://redis.io

1169:M 09 May 2021 14:53:38.636 # Server initialized
1169:M 09 May 2021 14:53:38.636 # 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.
1169:M 09 May 2021 14:53:38.637 * Loading RDB produced by version 6.2.3
1169:M 09 May 2021 14:53:38.637 * RDB age 48 seconds
1169:M 09 May 2021 14:53:38.637 * RDB memory usage when created 0.51 Mb
1169:M 09 May 2021 14:53:38.637 * DB loaded from disk: 0.000 seconds
1169:M 09 May 2021 14:53:38.637 * Ready to accept connections
```

7. Para ejecutar el cliente en otra terminal ejecutamos

```
src/redis-cli
```

```
ubuntu@INNOSOFT2: ~  
ubuntu@INNOSOFT2:~$ redis-cli  
127.0.0.1:6379> _
```

## CONFIGURACIÓN

Por defecto Redis utiliza el puerto 6379. La ruta por defecto de la configuración de redis es: `etc/redis/redis.conf`, donde se pueden configurar o cambiar algunos parámetros del servidor de acuerdo a la necesidad.

```
ubuntu@INNOSOFT2:/etc  
# redis.conf  
# If you are sure you want your instance to listen to all the interfaces  
# just comment out the following line:  
# bind 127.0.0.1 -:::1  
#  
# Protected mode is a layer of security protection, in order to avoid that  
# Redis instances left open on the internet are accessed and exploited.  
#  
# When protected mode is on and if:  
#  
# 1) The server is not binding explicitly to a set of addresses using the  
#    "bind" directive,  
# 2) No password is configured,  
#  
# The server only accepts connections from clients connecting from the  
# IPv4 and IPv6 loopback addresses 127.0.0.1 and ::1, and from Unix domain  
# sockets.  
#  
# By default protected mode is enabled. You should disable it only if  
# you are sure you want clients from other hosts to connect to Redis  
# even if no authentication is configured, nor a specific set of interfaces  
# are explicitly listed using the "bind" directive.  
protected-mode yes  
#  
# 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.  
#  
# In high requests-per-second environments you need a high backlog in order  
# to avoid slow clients connection issues. Note that the Linux kernel  
# will silently truncate it to the value of /proc/sys/net/core/somaxconn so  
# make sure to raise both the value of somaxconn and tcp_max_syn_backlog  
# in order to get the desired effect.  
tcp-backlog 511  
#  
# Unix socket.  
#  
# Specify the path for the unix socket that will be used to listen for  
# incoming connections. There is no default, so Redis will not listen  
# on a unix socket when not specified.  
# unixsocket /run/redis.sock  
# unixsocketperm 700  
#  
# Close the connection after a client is idle for N seconds (0 to disable)
```

También podemos ver la configuración de Redis desde el cliente (`redis-cli`) mediante los siguientes comandos:

### ***config get dir***

Devuelve el directorio donde se guardará la base de datos.

## ***config get dbfilename***

Devuelve el nombre de la base de datos.

Para cambiar un parámetro de configuración utilizamos el comando ***config set***

Ejemplo:

## ***config set dbfilename hotel.rdb***

Con lo anterior hemos cambiado el nombre de la base de datos a **hotel.rdb**

Si queremos listar todos los parámetros de Redis ejecutamos:

## ***config get \****



```
127.0.0.1:6379> config get *
1) "rdbchecksum"
2) "yes"
3) "daemonize"
4) "no"
5) "io-threads-do-reads"
6) "no"
7) "lua-replicate-commands"
8) "yes"
9) "always-show-logo"
10) "no"
11) "protected-mode"
12) "yes"
13) "rdbcompression"
14) "yes"
15) "rdb-del-sync-files"
16) "no"
17) "activeremhashing"
18) "yes"
19) "stop-writes-on-bgsave-error"
20) "yes"
21) "set-proc-title"
22) "yes"
23) "dynamic-hz"
24) "yes"
25) "lazyfree-lazy-eviction"
26) "no"
27) "lazyfree-lazy-expire"
28) "no"
29) "lazyfree-lazy-server-del"
30) "no"
31) "lazyfree-lazy-user-del"
32) "no"
33) "lazyfree-lazy-user-flush"
34) "no"
35) "repl-disable-tcp-nodelay"
36) "no"
37) "repl-diskless-sync"
38) "no"
39) "gopher-enabled"
40) "no"
41) "aof-rewrite-incremental-fsync"
42) "yes"
43) "no-appendfsync-on-rewrite"
44) "no"
45) "cluster-require-full-coverage"
46) "yes"
47) "rdb-save-incremental-fsync"
48) "yes"
```

Si queremos iniciar Redis por un puerto diferente al por defecto podemos ejecutar el siguiente comando al lanzar redis

## ***redis-server --port 6380***

Donde 6380 es el nuevo puerto por el que queremos lanzar el servicio redis.