«Cómo instalar y asegurar Redis en Ubuntu 18.04» por Mark Drake (versión anterior por Justin Ellingwood)

Introducción

Redis logotipo (https://en.wikipedia.org/wiki/File:Redis_Logo.svg)

Redis es una base de datos orientada a clave-valor conocida por su flexibilidad, desempeño y amplio soporte a varios lenguajes de programación. Este tutorial demostrará cómo instalar, configurar y asegurar Redis en un servidor con Ubuntu 18.04.

Prerrequisitos

Para completar esta guía usted debe tener acceso a un servidor Ubuntu 18.04 con un usuario *no raíz* pero con privilegios para ejecutar el comando **sudo** y un cortafuegos configurado de manera básica. Para esto, puede seguir el tutorial «Configuración inicial del servidor con el sistema Ubuntu 18.04».

Cuando esté listo para comenzar, conecte e ingrese sus credenciales a su servidor Ubuntu 18.04 como un usuario tipo *sudo* y ejecute las siguientes instrucciones.

Paso 1 — Instalando y configurando Redis

Para obtener la última versión de Redis, usaremos el comando **apt** para instalarlo desde los repositorios oficiales de Ubuntu.

Actualice su lista de paquetes **apt** e instale Redis por medio de:

```
$ sudo apt update
$ sudo apt install redis-server
```

Estas líneas descargarán e instalarán Redis y sus librerías. A continuación, hay un cambio en el fichero de configuración que es muy importante, dicho archivo fue generado automáticamente durante la instalación.

Abra este fichero con su editor de texto preferido, acá usaremos **nano**:

```
$ sudo nano /etc/redis/redis.conf
```

Dentro de este archivo localice la directiva **supervised**. Esta le permite declarar un sistema de inicio que administre a Redis como un servicio, proporcionando de esta manera mayor control sobre esta operación. La directiva **supervised** está configurada a **no** por defecto. Debido a que Ubuntu utiliza el sistema de inicio llamado **systemd**, cambie esta directiva de la siguiente manera:

```
. . .

# If you run Redis from upstart or systemd, Redis can interact with your
# supervision tree. Options:
# supervised no - no supervision interaction
# supervised upstart - signal upstart by putting Redis into SIGSTOP mode
# supervised systemd - signal systemd by writing READY=1 to $NOTIFY_SOCKE
T
# supervised auto - detect upstart or systemd method based on
```

```
# UPSTART_JOB or NOTIFY_SOCKET environment variables
# Note: these supervision methods only signal "process is ready."
# They do not enable continuous liveness pings back to your supervisor.
supervised systemd

. . .
```

En este punto, ese es el único cambio necesario a realizar en el archivo de configuración de Redis, así que guarde y cierre cuando haya finalizado. Luego reinicice el servicio Redis para que tome el cambio realizado a dicho fichero:

```
$ sudo systemctl restart redis.service
```

Con esto usted habrá instalado y configurado Redis y este software estará corriendo en su máquina. Antes de comenzar a usarlo es prudente verificar que Redis está trabajando de manera correcta.

Paso 2 — Comprobando a Redis

Como todo software recién instalado, es buena idea el asegurarse que Redis está funcionando como se espera antes de realizar cualquier otro cambio. En este paso iremos tras una serie de métodos útiles para verificar que Redis está trabajando correctamente.

```
$ sudo systemctl status redis
```

Si está corriendo sin error alguno, este comando mostrará una salida por pantalla similar a la siguiente:

```
 redis-server.service - Advanced key-value store
   Loaded: loaded (/lib/systemd/system/redis-
server.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2018-06-27 18:48:52 UTC; 12s ago
     Docs: http://redis.io/documentation,
```

```
          man:redis-server(1)
  Process: 2421 ExecStop=/bin/kill -s TERM $MAINPID (code=exited, status=
0/SUCCESS)
  Process: 2424 ExecStart=/usr/bin/redis-
server /etc/redis/redis.conf (code=exited, status=0/SUCCESS)
 Main PID: 2445 (redis-server)
    Tasks: 4 (limit: 4704)
   CGroup: /system.slice/redis-server.service
           ??2445 /usr/bin/redis-server 127.0.0.1:6379
. . .
```

Acá usted puede ver que Redis está funcionando y listo, lo que quiere decir que está ajustado para comenzar cada vez que el servidor inicie.

Esta configuración es la deseable para la mayoría de casos donde se utiliza Redis. Sin embargo si usted prefiere iniciar manualmente cada vez que reinicie su servidor puede configurar esto con el siguiente comando:

```
$ sudo systemctl disable redis
```

Para probar que Redis está funcionando de manera correcta, conéctese al servidor usando la interfaz de línea de comando:

```
$ redis-cli
```

En el indicador de comando pruebe la conectividad con **ping**:

```
127.0.0.1:6379> ping
```

Salida:

```
PONG
```

Esta salida confirma que la conexión con el servidor todavía está *viva.* A continuación verifique que sea capaz de crear alguna clave-valor corriendo lo siguiente:

```
127.0.0.1:6379> set test "It's working!"
```

Salida

```
OK
```

Ahora obtenga el valor tecleando lo siguiente:

```
127.0.0.1:6379> get test
```

Salida:

```
"It's working!"
```

Después de que se haya asegurado de que puede obtener el valor, salga del indicador de la línea de comandos y regrese al *shell* del servidor con:

```
127.0.0.1:6379> exit
```

Como comprobación final, revisaremos que Redis sea capaz de persistir aun cuando sea detenido o reiniciado. Para esto primero reinicie la instancia:

```
$ sudo systemctl restart redis
```

Luego conecte de nuevo a la línea de comando de Redis y confirme que el valro de prueba está aún disponible:

```
# redis-cli
```

```
127.0.0.1:6379> get test
```

Salida:

```
"It's working!"
```

Salga, otra vez, al *shell* del servidor:

```
127.0.0.1:6379> exit
```

Con todo esto realizado, su instalación de Redis está completamente operativa y lista para su uso. Sin embargo, algunos de sus ajustes de configuración predeterminados son inseguros y brindan a cualquiera actor malicioso las oportunidades para atacar y obtener acceso a vuestro servidor y sus datos. Los pasos restantes en este tutorial cubren métodos para mitigar estas vulnerabilidades, según lo establecido por el sitio web oficial de Redis. Aunque estos pasos son opcionales y Redis seguirá funcionando si elige no seguirlos, se recomienda encarecidamente que los complete para fortalecer la seguridad de su sistema.

Paso 3 — Enlazando al anfitrión local

Por defecto Redis es únicamente accesible desde el *anfitrión local* o **localhost**. No obstante si usted ha instalado y configurado Redis siguiendo el consejo de otros tutoriales, puede ser que lo tenga puesto a punto para recibir y permitir conexiones de cualquier lugar. Esto descrito no es tan

seguro como *enlazarlo al **localhost***.

Para corregir esto, abra el archivo de configuración de Redis:

```
$ sudo nano /etc/redis/redis.conf
```

Localice la línea que contenga el siguiente valor y asegúrese de desmarcar como comentario (ose, remueva el carácter "#" si existe al inicio de la línea):

```
bind 127.0.0.1 ::1
```

Guarde y cierre el archivo cuando haya terminado (si está utilizando **nano** presione CTRL + X y luego INTRO).

Ahora reinicie el servicio y confirme que **systemd** lea los cambios:

```
$ sudo systemctl restart redis
```

Verificar si este cambio ha tomado efecto implica ejecutar el siguiente comando llamado **netestat** acompañado de **grep**:

```
sudo netstat -lnp | grep redis
```

Salida:

```
tcp        0      0 127.0.0.1:6379          0.0.0.0:*               LISTE
N      14222/redis-server   tcp6       0      0 ::1:6379                :
::*                     LISTEN      14222/redis-server
```

Esta salida por pantalla demuestra que está enlazado al **localhost**, lo que dificulta en

sobremanera a cualquier atacante a realizar consultas o lograr acceso a su servidor. Sin embargo a estas alturas Redis todavía no está capacitado para exigir a los usuarios el autenticarse a sí mismos antes de modificar la configuración o los datos que alberga. Para remediar esto, Redis le permite requerir a los usuarios el autenticarse con una contraseña antes de hacer cambio alguno por medio del *cliente Redis* (**redis-cli**)**.**

Paso 4 — Configurando una contraseña para Redis

La configuración de una contraseña para Redis habilita para una de las dos características incrustadas de seguridad: el comando **auth**, el cual solicita a los clientes realizar una autenticación para acceder a la base de datos. La contraseña está configurada directamente en el archivo de configuración de Redis, **/etc/redis/redis.conf** así que abra de nuevo con su editor de texto preferido (aquí seguiremos usando **nano**):

```
$ sudo nano /etc/redis/redis.conf
```

Avance hasta la sección **SECURITY** y busque la directiva que está marcada como comentario:

```
# requirepass foobared
```

Removiendo el carácter # que está al inicio impide que la línea sea considerada un comentario, cambie la palabra **foobared** por una contraseña segura.

Justo arriba de la directiva **requirepass** en el fichero **redis.conf** hay un comentario que contiene una advertencia:

```
# Warning: since Redis is pretty fast an outside user can try up to
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
#
```

«Advertencia: debido a que Redi es muy rápido un usuario externo puede intentar entrar probando hasta 150 mil contraseñas por segundo. Esto significa que usted debería usar una contraseña extraordinariamente segura o sino será muy fácil de adivinar.»

Así que es importante que usted especifique una contraseña muy segura y tan larga como le sea posible. En vez de hacer una contraseña usted mismo, mejor utilice el comando **openssl** para generar una de manera aleatoria como en el siguiente ejemplo:

```
$ openssl rand 60 | openssl base64 -A
```

Debería ver algo parecido a esto:

Salida:

```
RBOJ9cCNoGCKhlEBwQLHri1g+atWgn4Xn4HwNUbtzoVxAYxkiYBi7aufl4MILv1nxBqR4L6NN
zI0X6cE
```

Después de copiar y pegar la salida del comando y establecerlo como nuevo valor a la directiva **requirepass**, debe quedar así la línea:

```
requirepass
RBOJ9cCNoGCKhlEBwQLHri1g+atWgn4Xn4HwNUbtzoVxAYxkiYBi7aufl4MILv1nxBqR4L6NN
zI0X6cE
```

Después de establecer la contraseña, guarde y cierre el fichero y reinicie a Redis:

```
$ sudo systemctl restart redis.service
```

Para confirmar que la contraseña sirve, entramos a la línea de comandos de Redis con:

```
redis-cli
```

Se muestra a continuación una secuencia de comandos para comprobar si funciona la contraseña asignada. El primer comando intenta establecer una clave a un valor antes de hacer la autenticación de usuario:

```
127.0.0.1:6379> set key1 10
```

Esto no funcionará porque no se ha autenticado como usuario, así que Redis responde con un error:

Salida:

```
(error) NOAUTH Authentication required.
```

Para autencarse como usuario usando la contraseña que fijamos en el archivo de configuración:

```
127.0.0.1:6379> auth la_contraseña_por_usted_fijada
```

Redis acusa recibo:

Salida:

```
OK
```

Ahora el correr el comando anterior otra vez, se realiza con éxito:

```
127.0.0.1:6379> set key1 10
```

Salida:

```
OK
```

Con la consulta **get key1** a Redis, este nos devuelve el valor de la nueva clave:

```
127.0.0.1:6379> get key1
```

Salida:

```
"10"
```

Veremos cómo renombrar los comandos de Redis para evitar, en el caso de que ya sea por error o por parte de algún atacante, se pueda causar daños serios a su máquina.

Paso 5 — Renombrando comandos peligrosos

La otra característica de seguridad integrada en Redis involucra renombrar o de manera completa ciertos comandos que son considerados peligrosos dado su alcance de uso.

Cuando son ejecutados por usuarios no autorizados, tales comandos puede ser usados reconfigurar, destruir o de otra manera desaparecer los datos. Como fue configurada la contraseña, renombrar o deshabilitar los comandos es configurado también en la misma sección **SECURITY** del fichero **/etc/redis/redis.conf**

Algunos de los comandos que son considerados peligrosos incluyen: **FLUSHDB**, **FLUSHALL**, **KEYS**, **PEXPIRE**, **DEL**, **CONFIG**, **SHUTDOWN**, **BGREWRITEAOF**, **BGSAVE**, **SAVE**, **SPOP**, **SREM**, **RENAME** y **DEBUG**. Esta no es una lista completa, pero renombrar o deshabilitar algunos o todos estos comandos de esa lista es un buen comienzo para fortalecer la seguridad del servidor que ejecuta Redis.

Si debe deshabilitar o cambiar el nombre de un comando depende de sus necesidades específicas o de las de su sitio. Si sabe que nunca usará un comando que pueda ser objeto de abuso, puede deshabilitarlo. De lo contrario, podría ser mejor cambiarle el nombre.

```
$ sudo nano /etc/redis/redis.conf
```

Advertencia: los siguientes pasos que muestran cómo deshabilitar y renombrar comandos *son meros ejemplos para usted*. Usted unicamente debería escoger el deshabilitar o renombrar los comandos que tengan sentido para usted. Puede revisar la lista completa de comandos y determinar por su cuenta cómo pueden tener un mal uso para su marco de trabajo.

Para deshabilitar un comando, simplemente renómbrelo a una cadena vacía (representada por un par de comillas sin ningún carácter entre ellas) tal como lo mostramos a continuación:

```
. . .
# It is also possible to completely kill a command by renaming it into
# an empty string:
#
rename-command FLUSHDB ""
rename-command FLUSHALL ""
rename-command DEBUG ""
. . .
```

Para renombrar un comando, asigne otro nombre como se muestra en el ejemplo. Los comandos que se renombran son difíciles de adivinar para otras personas, pero deben ser fáciles de memorizar para usted:

```
. . .
# rename-command CONFIG ""
rename-command SHUTDOWN SHUTDOWN_MENOT
rename-command CONFIG ASC12_CONFIG
. . .
```

Guarde los cambios y cierre el fichero **/etc/redis/redis.conf**

Después de renombrar un comando, aplique los cambios reiniciando a Redis:

```
$ sudo systemctl restart redis.service
```

Para probar las nuevas órdenes, entre en la línea de comandos de Redis con:

```
$ redis-cli
```

Una vez allí, autentíquese como usuario:

```
127.0.0.1:6379> auth la_contraseña_por_usted_fijada
```

Salida:

```
OK
```

Tomemos el caso que usted haya renombrado el comando **CONFIG** a **ASC12_CONFIG**, segú el ejemplo propuesto anteriormente. Primero trate de usar el comando original, este debería fallar en su cometido ya que fue renombrado:

```
127.0.0.1:6379> config get requirepass
```

Salida:

```
(error) ERR unknown command 'config'
```

Haciendo uso del comando renombrado será exitoso, a pesar del uso indistinto de mayúsculas y/o

minúsculas:

```
127.0.0.1:6379> asc12_config get requirepass
```

Finalmente salga de **redis-cli**:

127.0.0.1:6379>exit

Note que si usted ya utilizó y autenticó por la línea de comando de Redis y luego lo reinicia, usted necesitará volverse a autenticar, de otra manera usted obtendrá de nuevo el siguiente mensaje de error:

Salida:

```
NOAUTH Authentication required.
```

Con respecto a la práctica de renombrar los comandos, existe una cláusula de precaución al final de la sección **SECURITY** en el archivo **/etc/redis/redis.conf**

> Please note that changing the name of commands that are logged into the AOF file or transmitted to slaves may cause problems.

> Por favor tome en cuenta que cambiar el nombre de los comandos que ya están grabados en el fichero AOF o que sean transmitidos a los *esclavos* pueden causar problemas e imprevistos.

El proyecto Redis seleccionó los términos "master" (amo) y "slave" (esclavo), mientras que DigitalOcean generalmente prefiere los términos "primary" (primario) y "secondary" (secundario). Para prevenir confusiones, aquí hemos escogido usar los mismos términos usados que en la documentación de Redis.

Esto significa que si los comandos renombrados no están en el AOF, o ya están en el AOF pero aún no han sido transmitido a los esclavos, pues no habrá problema alguno.

Mantenga esto siempre presente cuando usted vaya a renombrar los comandos. El mejor momento para hacerlo es cuando usted **no está utilizando** el AOF persistente, o justo después de la instalación, osea, antes de que su aplicación Redis (y sus elementos) haya sido instalada.

Cuando usted está utilizando AOF y tiene un esquema de *amo-servidor* revise esta respuesta hecha en la pa?ina de inconvenientes del proyecto Redis. La siguiente cita es la respuesta hecha al usuario:

> The commands are logged to the AOF and replicated to the slave the same way they are sent, so if you try to replay the AOF on an instance that doesn't have the same renaming, you may face inconsistencies as the command cannot be executed (same for slaves).

> Los comandos que fueron grabados en el AOF y replicados a los esclavos son enviados así mismos, de la misma manera, así que si usted trata de reproducir dicho AOF en una instancia que **no tenga el mismo renombrado** pues usted enfrentará inconsistencias pues dicho comando no podrá ser ejecutado de la misma manera (e igualmente en los esclavos)

Dicho esto, la mejor manera de manejar los comandos renombrados es asegurarse priemro que dichos comandos sean también renombrados en **en todas las instancias** instaladas en el esquema *amo-esclavo*.

Conclusión

En este tutorial, usted instaló y configuró Redis, verificó que su instalación funciona correctamente y utilizó sus funciones de seguridad integradas para que sea menos vulnerable a los ataques maliciosos.

Tenga en cuenta que una vez que alguien inicia sesión en su servidor, es muy fácil eludir las características de seguridad específicas de Redis que hemos implementado. Por lo tanto, la característica de seguridad más importante en su servidor Redis es su muro de fuego (que configuró si siguió el prerrequisito tutorial de configuración inicial del servidor), ya que esto hace que sea extremadamente difícil para los actores maliciosos saltar esa barrera.

Escrito por: Mark Drake (versión anterior Justin Ellingwood).

Traducido por: Jimmy Olano

-

- **En castellano:** Este artículo es una traducción del inglés al castellano, publicado bajo licencia (en idioma inglés) «Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) » escrito por Mark Drake (una versión previa de este tutorial fue escrito por Justin Ellingwood @jmellingwood), publicado en línea por la empresa de alojamiento de máquinas virtuales DigitalOcean. El título original en idioma inglés es «***How To Install and Secure Redis on Ubuntu 18.04***» y hemos creado una copia en Wayback Machine para prevenir un posible enlace roto a futuro.