

Redis 的安装与配置

升级本地的 apt package cache:

```
$ sudo apt update
```

安装 Redis:

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

使用编辑器(vim or nano) 打开 *redis.conf* :

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

找到 `supervised` 指令并将其值修改为 `systemd`.

```
# 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
#                       requires "expect stop" in your upstart job config
#   supervised systemd - signal systemd by writing READY=1 to $NOTIFY_SOCKET
#                       on startup, and updating Redis status on a regular
#                       basis.
#   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 pings back to your supervisor.
#
# The default is "no". To run under upstart/systemd, you can simply uncomment
# the line below:
#
supervised systemd
```

保存文件并退出编辑器，重启 Redis 服务让修改的指令生效:

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

测试 Redis

检查 Redis 服务是否正常运行

```
sudo systemctl status redis
```

若 Redis 服务正常运行，会输出以下类似的消息：

```
● redis-server.service - Advanced key-value store
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-02-20 15:05:25 CST; 13min ago
     Docs: http://redis.io/documentation,
            man:redis-server(1)
  Main PID: 361767 (redis-server)
    Status: "Ready to accept connections"
     Tasks: 5 (limit: 18793)
    Memory: 4.6M
       CPU: 741ms
    CGroup: /system.slice/redis-server.service
            └─361767 /usr/bin/redis-server 127.0.0.1:6379
```

当看到 Redis 的状态是 **active (running)** 及 **enabled**，说明该服务会随操作系统而启动。

使用 redis-cli 测试 Redis 的功能是否正常：

```
$ redis-cli
```

使用 **ping** 测试 客户端与Redis 服务的连接。

```
→ ~ redis-cli
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> 
```

检查set 指令是否正常：

```
127.0.0.1:6379> set test "hello world!"
OK
127.0.0.1:6379> get test
"hello world!"
```

上述检查确认无误后，退出 **redis-cli**：

```
127.0.0.1:6379> exit
→ ~ 
```

检查 Redis 在停止服务或重启的情况下能否正常保存数据。

重新启动 Redis 实例：

```
$ sudo systemctl restart redis
```

再次用 **redis-cli** 连接到服务：

```
$ redis-cli
```

用 **get** 指令测试取值：

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

```
→ ~ sudo systemctl restart redis
[sudo] password for matrix:
→ ~ redis-cli
127.0.0.1:6379> get test
"hello world!"
127.0.0.1:6379> exit
```

完成测试后，可退出 `redis-cli`:

```
127.0.0.1:6379> exit
```

绑定到 localhost (可选)

默认情况下，Redis 只允许本地访问，若由于某些情况修改了配置，可参考以下方法重新绑定为本地访问。

打开 `redis.conf`:

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

定位到下面的一行并取消其注释：

```
bind 127.0.0.1 ::1
```

完成后保存文件并退出。

重新启动 Redis 服务，确保修改生效:

```
$ sudo systemctl restart redis
```

运行 `netstat` 检查修改是否生效:

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

```
→ ~ sudo netstat -lnp | grep redis
tcp        0      0 127.0.0.1:6379          0.0.0.0:*               LISTEN      356193/redis-server
tcp6       0      0 :::6379                 :::*                     LISTEN      356193/redis-server
```

以上输出说明 `redis-server` 已经被绑定到 `localhost(127.0.0.1)`，若有其他的 IP 出现，需要再检查修改是否成功，并重新启动 `redis` 服务。

注意：若系统中没有 `netstat`，可使用以下指令安装

```
sudo apt install net-tools
```

为 Redis 配置访问密码 (可选)

打开 `redis.conf`:

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

找到 `SECURITY` 部分或 `requirepass` 指令：

```
...  
# requirepass foobared  
...
```

移走 `#`，把 `foobared` 改为一个安全的密码。

可使用 `openssl` 命令生成一个随机密码:

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

```
→ ~ openssl rand 60 | openssl base64 -A  
5oeC9KTZ6cL1QWYskhjLnW1aS/7xFMM5Xn8uYALJNpfJRZrks2o0Kt2RASMPkLVVEs+Ea8Pma/CUCB9o%
```

把新生成的随机密码复制粘贴到 `requirepass` 后替换 `foobared`.
`/etc/redis/redis.conf`

```
# ...  
#  
requirepass 5oeC9KTZ6cL1QWYskhjLnW1aS/7xFMM5Xn8uYALJNpfJRZrks2o0Kt2RASMPkLVVEs+Ea8Pma/CUCB9o%  
# ...
```

设置密码后，重启 Redis 服务:

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

检查密码是否正常：

使用 `redis-cli` 连接到 Redis 服务

```
$ redis-cli
```

使用 `set` 指令测试

```
127.0.0.1:6379> set test_01 "test password"
```

若设置的密码生效，会提示 `(error) NOAUTH Authentication required.`

使用 `auth` 指令 进行与授权验证

```
127.0.0.1:6379> auth "密码"
```

若成功会提示 `OK`。
重新使用 set 指令测试

```
127.0.0.1:6379> set test_01 "test password"
```

使用 get 指令测试:

```
127.0.0.1:6379> get test_01
```

正常情况下, 能看到 test_01 的值。

```
→ ~ sudo systemctl restart redis.service
[sudo] password for matrix:
→ ~ redis-cli
127.0.0.1:6379> set test_01 "tesing password"
(error) NOAUTH Authentication required.
127.0.0.1:6379> auth 50ec9KTZ6cL1QWYskhjLnW1aS/7xFMMSXn8uYALJNpfJRZks2o0Kt2RASMPkLVVEs+Ea8Pma/CUCB9o%
OK
127.0.0.1:6379> set test_01 "tesing password"
OK
127.0.0.1:6379> get test_01
"tesing password"
```

安装 fastify-redis

在项目根目录下用命令 `npm ls fastify` 查看当前 fastify 的版本号:

```
$npm ls fastify
```

```
└─ fastify@3.25.3
   └─ fastify-cli@2.14.0
      └─ fastify@3.25.3 deduped
```

当前使用的版本是 `3.25.3`, 发布于 29 Dec, 2021, 所以需安装符合当前 Fastify 版本的 Fastify-redis.

3.25.3	29 Dec, 2021	0	C	0	H	1	M	0	L
3.25.2	23 Dec, 2021	0	C	0	H	1	M	0	L
3.25.1	21 Dec, 2021	0	C	0	H	1	M	0	L

出自 <https://security.snyk.io/package/npm/fastify>

故选择 `fastify-redis` 的 `V5.0.0` 版本。
在 InstaBIS 中安装 `fastify-redis`:

```
$ npm i @fastify/redis@5.0.0
```

命令汇总

```
sudo apt update //升级本地 apt package cache  
sudo apt install redis-server // 安装 Redis  
sudo vim /etc/redis/redis.conf // 编辑redis 配置文件  
sudo systemctl restart redis.service // 重启 redis 服务  
sudo systemctl status redis //查看 redis 运行状态  
redis-cli // 客户端工具
```

参考

[Getting started with Redis](#)

[How To Install and Secure Redis on Ubuntu 20.04](#)

[How to Install Redis on Ubuntu 20.04 / 18.04](#)

[fastify-redis](#)