安装参考：

<https://mp.weixin.qq.com/s/-IzRvIMjQjaGRpaR-wG1PA>

1. 下载：

访问https://redis.io/download地址到官网进行下载

1. 解压

执行tar -xvf

1. 将文件夹redis-7.2.4 重命名为redis

4、redis编译安装

1. redis编译，执行make命令。

出现：make[1]: Leaving directory '/usr/local/Redis/redis/src'

1. redis安装执行make install命令。

[root@iZ7xve0g1gz4tte7oyrzgiZ redis]# make install

cd src && make install

make[1]: Entering directory '/usr/local/Redis/redis/src'

CC Makefile.dep

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

INSTALL redis-server

INSTALL redis-benchmark

INSTALL redis-cli

make[1]: Leaving directory '/usr/local/Redis/redis/src'

5、redis启动

为了方便管理，将Redis目录中的conf配置文件和常用命令移动到统一目录中。

切换至redis根目录，创建bin和etc文件，将redis.conf文件移动到/usr/local/redis/etc/目录，将mkreleasehdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server 文件移动到/usr/local/redis/bin/目录。

1. 设置自启动

首先，新建一个系统服务文件：

vi /etc/systemd/system/redis.service

内容如下：(注意这里的内容根据实际的目录填写)

[Unit]

Description=Redis

After=network.target

[Service]

ExecStart=/usr/local/bin/redis-server /usr/local/redis/redis.conf --daemonize no

ExecStop=/usr/local/bin/redis-cli -h 8.138.26.188 -p 6379 shutdown

[Install]

WantedBy=multi-user.target

然后重载服务

systemctl daemon-reload

现在 redis 还没有实现开机自启，它只是被代理了，可以通过下面的命令启动 redis

systemctl stop redis

systemctl start redis

查看 redis 状态：

systemctl status redis

设置 redis 开机自启：

systemctl enable redis

这个命令是在Linux系统中启用Redis服务。它创建了一个符号链接，使得在系统启动时自动启动Redis服务。

1. 关闭自启动的操作

要关闭 Redis 服务的自启动，你可以使用 systemctl 命令来禁用它。以下是如何操作的步骤：

停止 Redis 服务：首先，你需要停止正在运行的 Redis 服务（如果它正在运行的话）。

systemctl stop redis

禁用 Redis 服务自启动：接下来，使用 disable 参数来禁止 Redis 服务在系统启动时自动启动。

systemctl disable redis

验证状态：你可以使用 is-enabled 参数来检查 Redis 服务是否已经被正确禁用。

systemctl is-enabled redis

这个命令将会返回一个状态，如果 Redis 服务已经被禁用，它将返回 disabled。

请注意，执行这些命令通常需要管理员权限，所以你可能需要在命令前加上 sudo，例如 sudo systemctl stop redis。

如果你在 systemctl 命令中遇到任何问题，可以检查 Redis 服务的状态和它的日志文件来获取更多信息。

1. 添加6379端口到公共区并永久生效

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

# 永久地添加一个服务或端口到public区域

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

# 重新加载配置以应用更改

sudo firewall-cmd --reload

1. 验证成功：

连接RedisDesktopManager



举例1：

Redis是一个高性能的内存数据库，用于存储和处理数据。它支持多种数据结构，如字符串、列表、集合、哈希表等，并提供了丰富的操作命令，可以满足各种场景下的数据存储和处理需求。

举个例子，假设你正在开发一个在线购物网站，需要实时地展示商品的库存信息。为了提高性能和响应速度，你可以使用Redis作为你的缓存服务器。

首先，你需要将商品库存信息存储在MySQL数据库中，然后使用Redis来缓存这些数据。当用户访问你的网站时，你可以先从Redis中读取缓存的库存信息，如果缓存中有数据，则直接返回给用户；如果缓存中没有数据，则需要从MySQL数据库中查询最新的库存信息，并将结果存储到Redis中，以便下次访问时可以直接读取缓存数据。

例如，当用户点击一个商品链接时，你可以先从Redis中读取该商品的库存信息，如果缓存中有数据，则直接显示给用户；如果缓存中没有数据，则需要从MySQL数据库中查询最新的库存信息，并将结果存储到Redis中，以便下次访问时可以直接读取缓存数据。

总之，Redis是一个非常强大的工具，可以帮助开发者轻松地实现数据的缓存和处理，提供高效可靠的服务给用户。

举例2：

假设你正在开发一个在线投票应用程序，需要实时地统计每个选项的得票数。为了提高性能和响应速度，你可以使用Redis作为你的计数器服务器。

首先，你需要将每个选项的初始得票数存储在MySQL数据库中，然后使用Redis来缓存这些数据。当用户进行投票时，你可以先从Redis中读取缓存的得票数，如果缓存中有数据，则直接更新该选项的得票数；如果缓存中没有数据，则需要从MySQL数据库中查询最新的得票数，并将结果存储到Redis中，以便下次访问时可以直接读取缓存数据。

例如，当用户点击一个选项进行投票时，你可以先从Redis中读取该选项的得票数，如果缓存中有数据，则直接更新该选项的得票数；如果缓存中没有数据，则需要从MySQL数据库中查询最新的得票数，并将结果存储到Redis中，以便下次访问时可以直接读取缓存数据。

总之，Redis是一个非常强大的工具，可以帮助开发者轻松地实现数据的计数和处理，提供高效可靠的服务给用户。

Redis运用：

Redis是一个在内存中存储数据的数据库，它特别快，就像是一个超级快速的储物柜，用来暂存数据。想象一下，你在写一篇很长的作文，但不希望每次都保存到硬盘上，因为那样会很慢而且容易丢失数据。相反，你会把正在写的内容暂时存放在一个快速的存储器里——这就是Redis的作用。

具体来说，比如你正在运营一个网上商城，每当有用户访问时，你都需要记录下用户的一些信息，比如他们查看了哪些商品、停留了多久等等。如果每次用户的这些行为都去查询数据库，可能会非常慢，影响用户体验。这时候，你可以把这些临时数据存在Redis里。由于Redis非常快，这样就能迅速响应用户的行为，提高网站运行效率。

总结一下，Redis就是用来快速存取那些临时需要频繁使用的数据，让软件跑得更快，用户体验更好。

好的，想象一下你在玩一个游戏，游戏里你有一个背包，这个背包就像Redis。

每次你找到宝物，比如金币或者宝石，你都会暂时放进背包里。这个背包特别神奇，因为你放入和取出东西都特别快，几乎一眨眼的功夫。但是，背包里的东西并不是永久保存的，如果你下线或者退出游戏，东西可能会消失（取决于Redis的配置，是不是把数据持久化到硬盘）。

现在，这个游戏里的其他玩家也想要交换物品。由于你的背包很快，你们可以迅速地交换彼此的物品，不用等待很长时间。这样，游戏体验就更流畅、更有趣了。

在现实世界中，这就相当于一个快速暂存服务，用来临时存放频繁使用的数据，使得数据交换、处理速度非常快。