



在Linux中安装Nginx、Tomcat和Mysql服务器

本篇文章将在Linux操作系统中安装Nginx、Tomcat和MySQL服务器，采用虚拟机Linux进行操作，以Ubuntu24.04为例进行安装。

Nginx

如果我们需要上传一个网站到服务器，需要安装HTTP Server（也称为Web服务器）。常用的Web服务器有nginx和Apache HTTP Server，当然我们常用的Tomcat也集成了Web服务的功能。

Nginx 是开源的轻量级 Web 服务器、反向代理服务器，以及负载均衡器和 HTTP 缓存器。其特点是高并发，高性能和低内存。

Nginx最初由Igor Sysoev为俄罗斯访问量第二的Rambler.ru站点开发，其将源代码以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、简单的配置文件和低系统资源的消耗而闻名。Nginx可以在大多数Unix、Linux OS上编译运行，并有Windows移植版。

安装Nginx

安装Nginx有几种常见的方法

方法一：通过APT包管理器安装，即使用 `apt-get` 命令安装

方法二：从源码包安装

通过 apt-get 安装 Nginx

使用以下命令进行安装：

```
sudo apt-get update
sudo apt-get install nginx
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo apt-get install nginx
[sudo] password for caicloudcat:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 55 not upgraded.
Need to get 552 kB of archives.
After this operation, 1,596 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.1 [31.2 kB]
Get:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.1 [521 kB]
Fetched 552 kB in 2s (225 kB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 191180 files and directories currently installed.)
Preparing to unpack .../nginx-common-1.24.0-2ubuntu7.1_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7.1) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx-1.24.0-2ubuntu7.1_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7.1) ...
Setting up nginx (1.24.0-2ubuntu7.1) ...
Setting up nginx-common (1.24.0-2ubuntu7.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

从源码包安装 Nginx

下载并解压源代码包

首先在本机去官网下载nginx包，安装稳定包

上传nginx到Linux系统自定义下的目录下（上传可以使用Xftp），并将其解压。

上面的一些初始步骤在之前的文章[专题8-压缩和解压缩文件](#)详细讲解过。

如果你要在Ubuntu中进行下载源代码包，则使用：

```
wget http://nginx.org/download/nginx-<version>.tar.gz
```

我安装的是稳定版本（stable version）是 1.26.2，所以命令应该是：

```
wget http://nginx.org/download/nginx-1.26.2.tar.gz
```

安装所需依赖包

Nginx是基于C语言开发的，所以安装nginx前需要安装C语言编译环境，HTTP rewrite 模块要求安装PCRE库，SSL模块要求OpenSSL库，执行以下命令来安装所需的依赖包

```
sudo apt-get install gcc libpcre3-dev zlib1g-dev openssl libssl-dev
```

那么如何查看已经安装的依赖包呢？

使用 `dpkg` 命令，该命令会列出所有已安装的软件包

```
dpkg -l
```

但是内容过长，不容易查看，因此可以使用 `grep` 来筛选特定的软件包：

```
dpkg -l | grep <package-name>
```

编译安装 Nginx

首先进入解压缩完后的nginx目录，我这里是 `nginx-1.26.2`：

```
cd nginx-1.26.2
```

进入nginx-1.26.2目录后，即可执行configure脚本生成编译配置文件Makefile：

```
./configure --prefix=/usr/local/nginx
```

```
Configuration summary
+ using system PCRE library
+ OpenSSL library is not used
+ using system zlib library

nginx path prefix: "/usr/local/nginx"
nginx binary file: "/usr/local/nginx/sbin/nginx"
nginx modules path: "/usr/local/nginx/modules"
nginx configuration prefix: "/usr/local/nginx/conf"
nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
nginx pid file: "/usr/local/nginx/logs/nginx.pid"
nginx error log file: "/usr/local/nginx/logs/error.log"
nginx http access log file: "/usr/local/nginx/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/nginx/nginx-1.26.2$
```

这是一个非常简易的配置安装，你可以配置你需要的module，完整的文档可参考[Building nginx from Sources](#)。



"注意"

`--prefix` 是配置安装的路径，如果不配置该选项，安装后可执行文件默认放在 `/usr/local/bin`，库文件默认放在 `/usr/local/lib`，配置文件默认放在 `/usr/local/etc`，其它的资源文件放在 `/usr/local/share`，比较凌乱。上述命令配置 `--prefix`，可以把所有资源文件放在 `/usr/local/nginx` 的路径中，不会杂乱。

稍微讲解上述的一些内容解释：

- `-prefix` 指定nginx安装目录
- `-pid-path` 指向nginx的pid
- `-lock-path` 锁定安装文件，防止被恶意篡改或误操作
- `-error-log` 错误日志
- `-http-log-path` http日志
- `-with-http_gzip_static_module` 启用gzip模块，在线实时压缩输出数据流
- `-http-client-body-temp-path` 设定客户端请求的临时目录
- `-http-proxy-temp-path` 设定http代理临时目录
- `-http-fastcgi-temp-path` 设定fastcgi临时目录
- `-http-uwsgi-temp-path` 设定uwsgi临时目录
- `-http-scgi-temp-path` 设定scgi临时目录

Makefile 是一个用于自动化编译和构建项目的文件，通常用于管理源代码的编译过程。它包含了一组规则，指示 `make` 工具如何从源文件生成目标文件（如可执行文件或库文件）。Makefile 使得项目的构建过程更加高效、可重复和自动化，特别是在大型项目中。

一个基本的 Makefile 包括以下部分：

1. 目标 (Target)：

目标是 Makefile 中需要生成的文件。通常是一个可执行文件、一个库文件或中间文件（例如 `.o` 文件）。目标通常出现在每行的最前面。

2. 依赖 (Dependencies) :

依赖是目标所依赖的文件。只有当依赖的文件发生变化时，目标才会重新构建。依赖关系描述了源文件和目标文件之间的关系。

3. 规则 (Rule) :

规则描述了如何从依赖生成目标。通常包括一个命令，这个命令会在目标需要更新时执行。规则通常由命令行组成，命令前面会有一个 tab（制表符）字符。

接下来进行编译并安装：

```
make # 编译源码
sudo make install # 安装Nginx
```

如果没有安装 `make`，则使用下面的命令进行安装：

```
sudo apt update
sudo apt install make
```

安装完成后，使用 `whereis` 命令查看是否安装成功：

```
whereis nginx
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/nginx/nginx-1.26.2$ whereis nginx
nginx: /usr/sbin/nginx /etc/nginx /usr/local/nginx /usr/share/nginx /usr/share/man/man8/nginx.8.g
z
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/nginx/nginx-1.26.2$
```

出现 `/usr/local/nginx` 即完成安装，其他的是之前使用apt-get方法安装Nginx留存的。

或者可以使用下面的命令，检查Nginx的版本：

```
nginx -V
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx/sbin$ nginx -V
nginx version: nginx/1.24.0 (Ubuntu)
built with OpenSSL 3.0.13 30 Jan 2024
TLS SNI support enabled
configure arguments: --with-cc-opt='-g -O2 -fno-omit-frame-pointer -mno-omit-leaf-frame-pointer -ffile-prefix-map=/build/nginx-DLMnQR/nginx-1.24.0=. -flto=auto -ffat-lto-objects -fstack-protector-strong -fstack-clash-protection -Wformat -Werror=format-security -fcf-protection -fdebug-prefix-map=/build/nginx-DLMnQR/nginx-1.24.0=/usr/src/nginx-1.24.0-2ubuntu7.1 -fPIC -Wdate-time -D_FORTIFY_SOURCE=3' --with-ld-opt='-Wl,-Bsymbolic-functions -flto=auto -ffat-lto-objects -Wl,-z,relro -Wl,-z,now -fPIC' --prefix=/usr/share/nginx --conf-path=/etc/nginx/nginx.conf --http-log-path=/var/log/nginx/access.log --error-log-path=stderr --lock-path=/var/lock/nginx.lock --pid-path=/run/nginx.pid --modules-path=/usr/lib/nginx/modules --http-client-body-temp-path=/var/lib/nginx/body --http-fastcgi-temp-path=/var/lib/nginx/fastcgi --http-proxy-temp-path=/var/lib/nginx/proxy --http-scgi-temp-path=/var/lib/nginx/scgi --http-uwsgi-temp-path=/var/lib/nginx/uwsgi --with-compat --with-debug --with-pcre-jit --with-http_ssl_module --with-http_stub_status_module --with-http_realip_module --with-http_auth_request_module --with-http_v2_module --with-http_dav_module --with-http_slice_module --with-threads --with-http_addition_module --with-http_flv_module --with-http_gunzip_module --with-http_gzip_static_module --with-http_mp4_module --with-http_random_index_module --with-http_secure_link_module --with-http_sub_module --with-mail_ssl_module --with-stream_ssl_module --with-stream_ssl_preread_module --with-stream_realip_module --with-http_geoip_module=dynamic --with-http_image_filter_module=dynamic
```

这是 `usr/local/nginx` 目录的详细内容：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/nginx/nginx-1.26.2$ cd /usr/local/nginx
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx$ ls -l
total 16
drwxr-xr-x 2 root root 4096 Dec  3 19:12 conf
drwxr-xr-x 2 root root 4096 Dec  3 19:12 html
drwxr-xr-x 2 root root 4096 Dec  3 19:12 logs
drwxr-xr-x 2 root root 4096 Dec  3 19:12 sbin
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx$
```

进入 `nginx` 目录下的 `sbin` 目录，执行命令：

```
sudo ./nginx # 启动
sudo ./nginx -s stop # 停止
sudo ./nginx -s reload # 重新加载
```

但在运行上面的代码时候，出现该问题：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx/sbin$ sudo ./nginx
[sudo] password for caicloudcat:
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] still could not bind()
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx/sbin$
```

原因是该端口已经有服务，因为我之前使用apt包安装Nginx后该服务还在启动中。

先将apt包安装的Nginx服务停掉，再试一遍：

```
sudo systemctl stop nginx
sudo lsof -i:80
sudo /usr/local/nginx/sbin/nginx
sudo lsof -i:80
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/lib/systemd/system$ sudo systemctl stop nginx
[sudo] password for caicloudcat:
caicloudcat@caicloudcat-VMware-Virtual-Platform:/lib/systemd/system$ sudo lsof -i:80
caicloudcat@caicloudcat-VMware-Virtual-Platform:/lib/systemd/system$ sudo /usr/local/nginx/sbin/nginx
caicloudcat@caicloudcat-VMware-Virtual-Platform:/lib/systemd/system$ sudo lsof -i:80
COMMAND  PID  USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
nginx    9021  root   6u  IPv4  40799      0t0  TCP *:http (LISTEN)
nginx    9022  nobody 6u  IPv4  40799      0t0  TCP *:http (LISTEN)
caicloudcat@caicloudcat-VMware-Virtual-Platform:/lib/systemd/system$
```

可以发现启动成功。

配置其他现有 Web 服务器

如果您的 Ubuntu 服务器上安装了其他 Web 服务器（例如 Apache），请在安装 Nginx 之前卸载它们。这将避免任何冲突或端口绑定问题。

```
sudo apt-get remove apache2
```

或者，如果您想与 Apache 一起运行 Nginx，您可以选择使用 Nginx 作为 Apache 的反向代理。此配置允许 Nginx 处理传入请求并将其转发给 Apache 进行处理。此设置可以提供两个 Web 服务器的优点。

测试Nginx

使用下面的命令来检查服务是否正在运行：

```
systemctl status nginx
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-11-26 21:02:50 CST; 4min 28s ago
     Docs: man:nginx(8)
   Process: 3798 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 3799 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 3801 (nginx)
    Tasks: 5 (limit: 4558)
   Memory: 3.7M (peak: 4.1M)
      CPU: 42ms
   CGroup: /system.slice/nginx.service
           └─3801 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─3802 "nginx: worker process"
               └─3803 "nginx: worker process"
                 └─3804 "nginx: worker process"
                   └─3805 "nginx: worker process"

Nov 26 21:02:50 caicloudcat-VMware-Virtual-Platform systemd[1]: Starting nginx.service - A high performance web server and a reverse p
Nov 26 21:02:50 caicloudcat-VMware-Virtual-Platform systemd[1]: Started nginx.service - A high performance web server and a reverse pr
lines 1-19/19 (END)
```

从上面的图片可以确认服务已经成功启动了。不过，最好的方式是实际从Nginx 要求一个网页。

你可以通过访问预设的Nginx 登陆页面，来确认软体运行正常。只需前往你服务器的IP 地址，可以通过 `ifconfig` 或者 `ip address` 来查看IP地址。

将它输入到你浏览器的地址栏中：

```
http://your_serve_ip
```

然后就会看到预设的Nginx 登陆页面：



提示：如果启动 Nginx 服务时出现错误，很有可能是 80 端口已被使用。Nginx 默认使用端口 80 进行 HTTP 流量。如果另一个服务已经使用了80端口，Nginx将无法启动。要检查80端口是否被使用，可以运行以下命令：

```
sudo lsof -i :80
```



```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo lsof -i :80
[sudo] password for caicloudcat:
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
nginx 1096 root 5u IPv4 23108 0t0 TCP *:http (LISTEN)
nginx 1096 root 6u IPv6 23109 0t0 TCP *:http (LISTEN)
nginx 1100 www-data 5u IPv4 23108 0t0 TCP *:http (LISTEN)
nginx 1100 www-data 6u IPv6 23109 0t0 TCP *:http (LISTEN)
nginx 1101 www-data 5u IPv4 23108 0t0 TCP *:http (LISTEN)
nginx 1101 www-data 6u IPv6 23109 0t0 TCP *:http (LISTEN)
nginx 1102 www-data 5u IPv4 23108 0t0 TCP *:http (LISTEN)
nginx 1102 www-data 6u IPv6 23109 0t0 TCP *:http (LISTEN)
nginx 1105 www-data 5u IPv4 23108 0t0 TCP *:http (LISTEN)
nginx 1105 www-data 6u IPv6 23109 0t0 TCP *:http (LISTEN)
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

这里看到是Nginx在运行服务，如果另一个服务使用端口 80，您可以停止该服务或将 Nginx 配置为使用其他端口。

配置防火墙

如果您已在系统上启用 UFW 防火墙，请确保对其进行适当配置，以允许 Nginx 使用的端口上的传入流量。Nginx 使用的默认端口是 HTTP 的 80 和 HTTPS 的 443。

首先使用 `sudo ufw status verbose` 来查看防火墙状态（包括规则详细信息）

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22/tcp ALLOW IN Anywhere
22/tcp (v6) ALLOW IN Anywhere (v6)
```

您可以运行以下命令来允许 Nginx 的流量。

```
sudo ufw allow 'Nginx HTTP'
sudo ufw allow 'Nginx HTTPS'
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo ufw allow 'Nginx HTTP'
Rule added
Rule added (v6)
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo ufw allow 'Nginx HTTPS'
Rule added
Rule added (v6)
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

再查看防火墙状态，可以看到如下：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22/tcp ALLOW IN Anywhere
80/tcp (Nginx HTTP) ALLOW IN Anywhere
443/tcp (Nginx HTTPS) ALLOW IN Anywhere
22/tcp (v6) ALLOW IN Anywhere (v6)
80/tcp (Nginx HTTP (v6)) ALLOW IN Anywhere (v6)
443/tcp (Nginx HTTPS (v6)) ALLOW IN Anywhere (v6)

caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

UFW (Uncomplicated Firewall) 是一个简单的防火墙管理工具，主要用于简化 Linux 系统上的 iptables 配置。它提供了一个简单的命令行接口，使得设置防火墙规则变得更加直观和易于管理。UFW 适用于大多数 Linux 发行版，如 Ubuntu 和 Debian。

管理Nginx进程 (Nginx Processing)

要停止你的网页伺服器，输入以下指令：

```
sudo systemctl stop nginx
```

如果网页伺服器停止了，要重新启动它，请输入以下指令：

```
sudo systemctl start nginx
```

如果要先停止再重新启动服务，请输入以下指令：

```
sudo systemctl restart nginx
```

如果你只是在进行设定更改，通常Nginx 可以在不断开连线的情况下重新载入。要这么做，输入以下指令：

```
sudo systemctl reload nginx
```

预设情况下，Nginx 设定为在伺服器启动时 **自动启动**。如果你不希望这样，可以通过输入以下

指令来停用这个功能：

```
sudo systemctl disable nginx
```

若要重新启用服务在开机时自动启动，请输入以下指令：

```
sudo systemctl enable nginx
```

那么为什么会有这些快捷指令？它是Nginx安装后自动配置的吗？其实是通过配置Systemd File 服务单元文件，来管理Nginx服务。

配置 Systemd File

通过 `apt-get` 来安装Nginx，它会自动在 `/lib/systemd/system` 目录下给你创建一个 `nginx.service`。

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/lib/systemd/system$ ls -l | grep nginx
-rw-r--r-- 1 root root 1083 Dec  1 2023 nginx.service
caicloudcat@caicloudcat-VMware-Virtual-Platform:/lib/systemd/system$
```

我们来看看它长什么样子：

```

# Stop dance for nginx
# =====
#
# ExecStop sends SIGQUIT (graceful stop) to the nginx process.
# If, after 5s (--retry QUIT/5) nginx is still running, systemd takes control
# and sends SIGTERM (fast shutdown) to the main process.
# After another 5s (TimeoutStopSec=5), and if nginx is alive, systemd sends
# SIGKILL to all the remaining processes in the process group (KillMode=mixed).
#
# nginx signals reference doc:
# http://nginx.org/en/docs/control.html
#
[Unit]
# Description提供了服务的简短描述
Description=A high performance web server and a reverse proxy server
# Documentation提供了该服务的相关文档：指向Nginx的手册页
Documentation=man:nginx(8)
# After定义了该服务的执行顺序，他将在 network-online.target、remote-fs.target、
# nss-lookup.target 完成后执行，也就是在网络配置、远程文件系统和 NSS（名称服务切换）准备好之后启动
After=network-online.target remote-fs.target nss-lookup.target
# Nginx 服务希望 network-online.target 处于激活状态，但它不是强制要求。
# 也就是，即使 network-online.target 没有完全完成，Nginx 仍会尝试启动。
Wants=network-online.target

[Service]
# 表示 Nginx 是通过派生子进程来运行的。
Type=forking
# PIDFile 指定了 Nginx 主进程的 PID 文件路径，该文件包含 Nginx 主进程的 PID
# systemd 会通过这个文件来管理 Nginx 的启动、停止等操作。
PIDFile=/run/nginx.pid
# ExecStartPre 命令在实际启动 Nginx 之前执行，用于检查配置文件是否正确。
# -t 用于测试配置，-q 用于减少输出，
# -g 'daemon on; master_process on;' 用于确保 Nginx 在正确的模式下运行。
ExecStartPre=/usr/sbin/nginx -t -q -g 'daemon on; master_process on;'
# 启动 Nginx 的实际命令
ExecStart=/usr/sbin/nginx -g 'daemon on; master_process on;'
# 该命令用于重新加载 Nginx 配置文件，而不需要停止 Nginx 服务。
ExecReload=/usr/sbin/nginx -g 'daemon on; master_process on;' -s reload
# ExecStop 用于停止 Nginx 服务。它向 Nginx 主进程发送 `QUIT` 信号，以优雅的方式关闭 Nginx。

```

```
# --retry QUIT/5 表示如果在 5 秒内 Nginx 进程没有退出，start-stop-daemon 会强制退出。
# - 符号表示即使命令执行失败，也不会导致服务的停止失败。
ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid
# TimeoutStopSec 指定了在发送 QUIT 信号后，等待 Nginx 停止的最大时间。
# 如果 Nginx 在 5 秒内没有停止，systemd 将强制停止 Nginx。
TimeoutStopSec=5
# KillMode 设置为 mixed，表示在停止服务时，systemd 会先发送 QUIT 信号给主进程（优雅停机）
# 然后，如果主进程在 TimeoutStopSec 时间内没有终止，systemd 会发送 SIGTERM 信号
# 最后发送 SIGKILL 强制终止所有相关进程。
KillMode=mixed

[Install]
# 指定了当系统进入 multi-user.target 时，Nginx 服务应该被自动启动。
WantedBy=multi-user.target
```

该配置文件为 Nginx 提供了一个完善的启动、停止和重载机制，确保服务能够在系统启动时自动管理。

如果你是使用源代码包进行安装，则可以自己创建来管理Nginx。

可以使用下面的命令创建：

```
sudo vim /lib/systemd/system/nginx.service
```

然后将相应的配置写入，这里不再详细介绍。

最后，重新加载：

```
sudo systemctl daemon-reload
```

从本地上传网站

我们接下来将从本地（Windows系统）上传一个网站到Nginx服务器，并测试。

配置文件知识

首先我们用apt包管理器下载的Nginx来查看一些基础的配置文件

进入 `/etc/nginx` 目录下:

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/nginx/sites-available$ cd /etc/nginx
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/nginx$ ls -l
total 68
drwxr-xr-x 2 root root 4096 Sep 10 21:27 conf.d
-rw-r--r-- 1 root root 1125 Dec 1 2023 fastcgi.conf
-rw-r--r-- 1 root root 1055 Dec 1 2023 fastcgi_params
-rw-r--r-- 1 root root 2837 Dec 1 2023 koi-utf
-rw-r--r-- 1 root root 2223 Dec 1 2023 koi-win
-rw-r--r-- 1 root root 5465 Dec 1 2023 mime.types
drwxr-xr-x 2 root root 4096 Sep 10 21:27 modules-available
drwxr-xr-x 2 root root 4096 Sep 10 21:27 modules-enabled
-rw-r--r-- 1 root root 1446 Dec 1 2023 nginx.conf
-rw-r--r-- 1 root root 180 Dec 1 2023 proxy_params
-rw-r--r-- 1 root root 636 Dec 1 2023 scgi_params
drwxr-xr-x 2 root root 4096 Nov 26 21:02 sites-available
drwxr-xr-x 2 root root 4096 Nov 26 21:02 sites-enabled
drwxr-xr-x 2 root root 4096 Nov 26 21:02 snippets
-rw-r--r-- 1 root root 664 Dec 1 2023 uwsgi_params
-rw-r--r-- 1 root root 3071 Dec 1 2023 win-utf
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/nginx$
```

进入 `sites-available` 目录下, 可以看到有一个 `default` 文件:

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/nginx$ cd sites-available
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/nginx/sites-available$ ls -l
total 4
-rw-r--r-- 1 root root 2412 Dec 1 2023 default
caicloudcat@caicloudcat-VMware-Virtual-Platform:/etc/nginx/sites-available$
```

我们用 `vim` 来看看 `default` 文件的内容:

```
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# 你应该查看以下 URL，以便对 Nginx 配置文件有深入的了解，从而充分释放 Nginx 的强大功能。
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
# 在大多数情况下，管理委员会从 sites-enabled/ 中删除此文件，
# 并将其作为 sites-available 内的参考，以便 nginx 打包团队继续更新它。
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
# 此文件将自动加载其他应用程序（例如 Drupal 或 Wordpress）提供的配置文件。
# 这些应用程序将在具有该包名称的路径下提供，例如 /drupal8。
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
```

```

# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $url $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
#location ~ /\.php$ {
#    include snippets/fastcgi-php.conf;
#
#    # With php-fpm (or other unix sockets):
#    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
#    # With php-cgi (or other tcp sockets):
#    fastcgi_pass 127.0.0.1:9000;
#}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}
}

# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#

```



```
#server {
#     listen 80;
#     listen [::]:80;
#
#     server_name example.com;
#
#     root /var/www/example.com;
#     index index.html;
#
#     location / {
#         try_files $uri $uri/ =404;
#     }
#}
```

接下来讲解一些上述内容：

server 块

Nginx 的 `server` 块用来定义一个虚拟主机（vhost），这意味着它负责处理来自客户端的请求。每个 `server` 块都能绑定一个 IP 地址和端口，可以有多个 `server` 块来处理不同的域名或 IP。

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
}
```

这行代码表示：

- `listen 80`：该服务器监听 IPv4 的 80 端口（HTTP 默认端口）。
- `listen [::]:80`：该服务器监听 IPv6 的 80 端口。
- `default_server`：这个配置会将该服务器设置为 **默认服务器**，当请求没有匹配到任何 `server_name` 时会使用这个服务器。

SSL配置

```
# SSL configuration
# listen 443 ssl default_server;
# listen [::]:443 ssl default_server;
# include snippets/snakeoil.conf;
```

这部分是 SSL 配置，但它被注释掉了，表示此服务器没有启用 HTTPS 配置。如果启用了，你需要提供 SSL 证书文件以及相关配置。443 是 HTTPS 的默认端口。

root 指令

```
root /var/www/html;
```

这行定义了网站的根目录。Nginx 会从这个路径查找文件并返回给客户端。当用户访问网站时，Nginx 会寻找相应的文件（如 `index.html`）并返回给用户。

index 指令

```
index index.html index.htm index.nginx-debian.html;
```

这指定了默认的文件列表，当访问根目录（`/`）时，Nginx 会优先返回这些文件。

如果用户访问网站的根目录而不指定文件名，Nginx 会依次查找 `index.html`、`index.htm` 和 `index.nginx-debian.html`。

server_name 指令

```
server_name _;
```

`server_name` 指令用来定义此服务器响应的域名。在这里，`_` 表示默认匹配任何请求，这在没有配置特定域名时会生效。通常你会把它替换为你自己的域名，例如 `example.com`。

location 块

```
location / {
    try_files $url $uri/ =404;
}
```

`location` 块用于匹配请求的 URI，并为其提供相应的处理规则。在这里：

- `try_files $url $uri/ =404`：Nginx 会首先检查请求的 URL 是否与某个文件匹配，如果匹配则返回该文件。如果没有匹配到文件，接着会检查是否有匹配的目录。如果都没有匹配，最后会返回 404 错误。

PHP 配置

```
#location ~ /\.php$ {  
#    include snippets/fastcgi-php.conf;  
#    fastcgi_pass unix:/run/php/php7.4-fpm.sock;  
#    fastcgi_pass 127.0.0.1:9000;  
#}
```

这部分被注释掉了，它是用来处理 PHP 请求的。如果你需要处理 PHP 文件（如 `.php` 文件），则需要配置 FastCGI 与 PHP-FPM。

.htaccess 文件的限制

```
#location ~ /\.ht {  
#    deny all;  
#}
```

这部分的配置用来阻止访问 `.htaccess` 文件。如果你使用 Apache 的 `.htaccess` 文件，Nginx 会拒绝对其的访问，因为 Nginx 并不使用 `.htaccess` 文件，而是通过配置文件直接进行访问控制。

在 `sites-available` 目录中有许多虚拟主机配置文件，通常会通过符号链接（symlink）将需要启用的配置链接到 `sites-enabled` 目录。

apt包管理器安装的Nginx

首先来到其默认的网页管理目录下：

```
cd /var/www/html  
ls -l
```

```
drwxr-xr-x 2 root root 4096 Nov 26 21:02 html
caicloudcat@caicloudcat-VMware-Virtual-Platform:/var/www$ cd /var/www/html
caicloudcat@caicloudcat-VMware-Virtual-Platform:/var/www/html$ ls -l
total 4
-rw-r--r-- 1 root root 615 Nov 26 21:02 index.nginx-debian.html
caicloudcat@caicloudcat-VMware-Virtual-Platform:/var/www/html$
```

可以看到该目录下有一个默认的 `index.nginx-debian.html` 网页，用 `vim` 查看该文件：

```
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

15,0-1

Bot

可以看到就是之前的页面的html源代码。

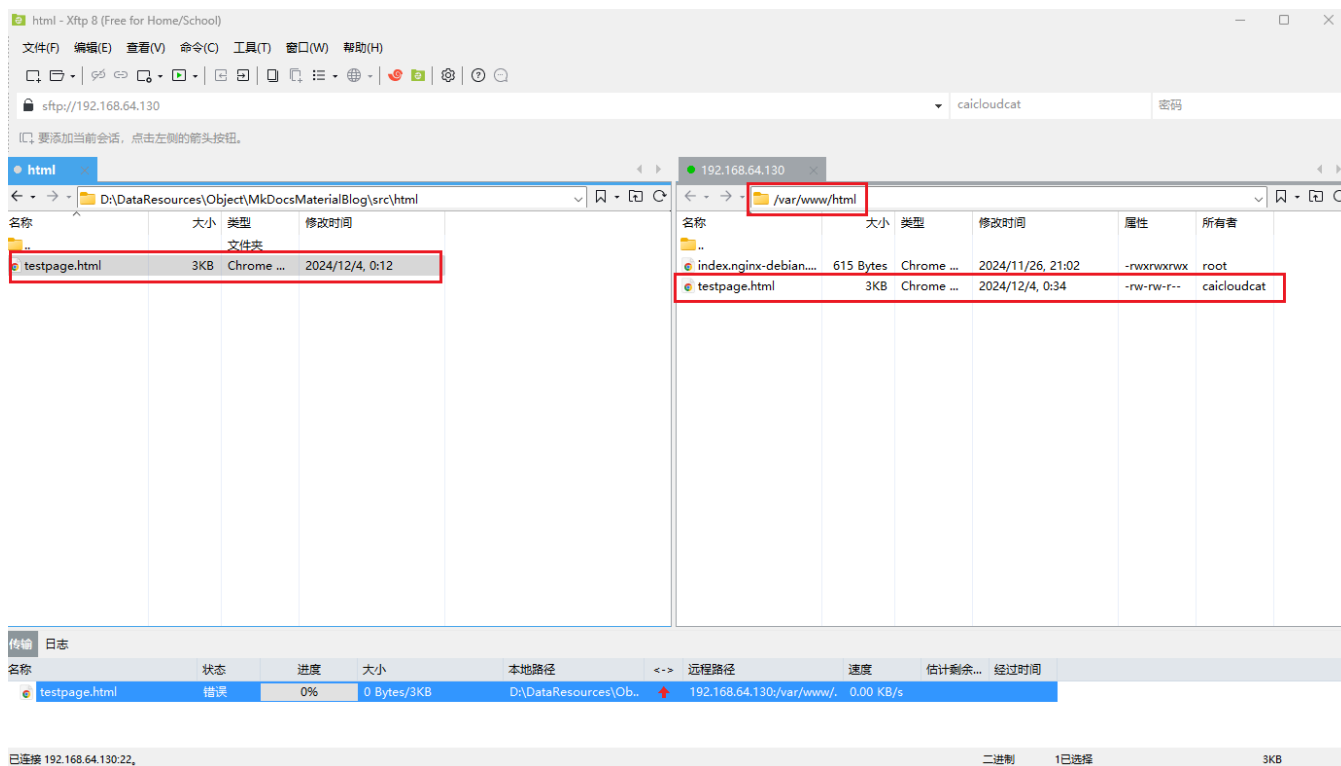
我们可以将自己本地的测试网站上传到该目录下，但是因为权限不够而无法上传。

`www` 和 `html` 目录的权限都是 `rwxr-xr-x`。

比较简单的方法就是修改该目录的权限：

```
sudo chmod -R 777 /var/www/
```

执行命令后在传输，即可成功



在网站中输入 `http://your_server_ip/testpage.html` 查看是否正确显示，在这里我的
是 `http://192.168.64.10/testpage.html`



源码包安装的Nginx

其步骤核心跟上面的大致相同，首先关闭来自于上面的Nginx服务，并启动当前的Nginx服务。

```
sudo systemctl stop nginx
sudo /usr/local/nginx/sbin/nginx
```

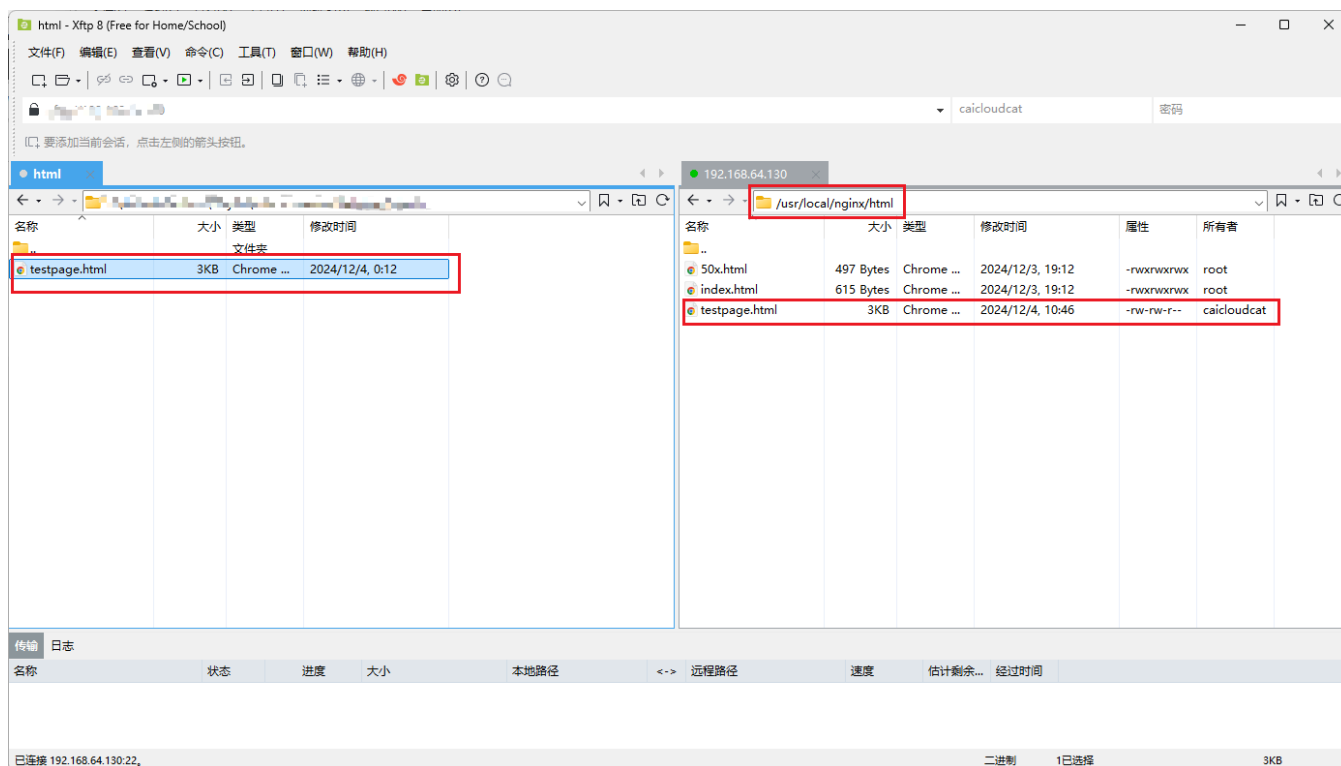
接着去安装Nginx的目录下，即 `/usr/local/nginx` 下。

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ cd /usr/local/nginx
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx$ ls -l
total 36
drwx----- 2 nobody root 4096 Dec  3 19:33 client_body_temp
drwxr-xr-x 2 root  root 4096 Dec  3 19:12 conf
drwx----- 2 nobody root 4096 Dec  3 19:33 fastcgi_temp
drwxr-xr-x 2 root  root 4096 Dec  3 19:12 html
drwxr-xr-x 2 root  root 4096 Dec  4 10:33 logs
drwx----- 2 nobody root 4096 Dec  3 19:33 proxy_temp
drwxr-xr-x 2 root  root 4096 Dec  3 19:12 sbin
drwx----- 2 nobody root 4096 Dec  3 19:33 scgi_temp
drwx----- 2 nobody root 4096 Dec  3 19:33 uwsgi_temp
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx$
```

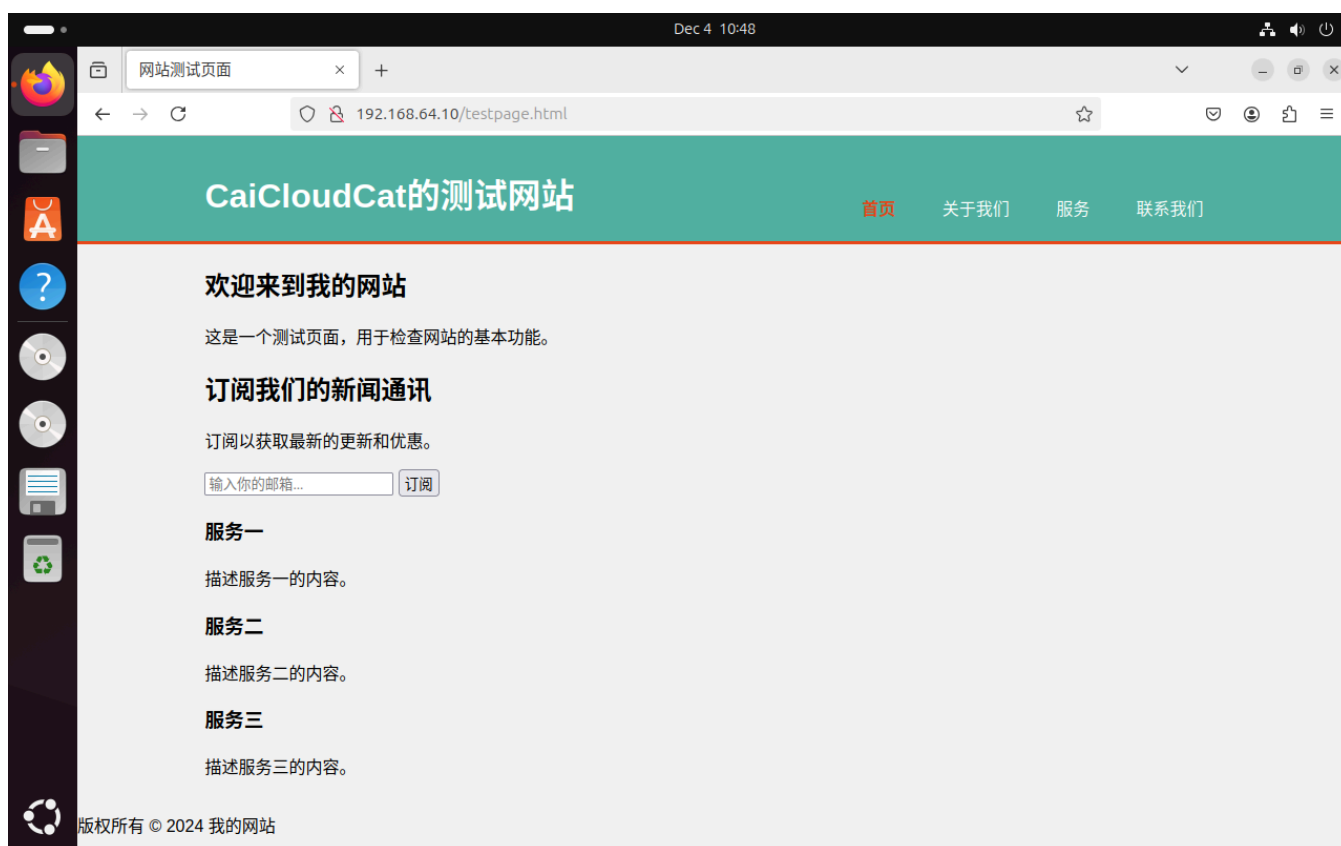
跟之前的一样，将 `html` 目录权限进行修改，以及上传我的测试网页。

```
sudo chmod -R 777 /usr/local/nginx/html
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx$ sudo chmod -R 777 html
caicloudcat@caicloudcat-VMware-Virtual-Platform:/usr/local/nginx$ ls -l
total 36
drwx----- 2 nobody root 4096 Dec  3 19:33 client_body_temp
drwxr-xr-x 2 root  root 4096 Dec  4 10:36 conf
drwx----- 2 nobody root 4096 Dec  3 19:33 fastcgi_temp
drwxrwxrwx 2 root  root 4096 Dec  3 19:12 html
drwxr-xr-x 2 root  root 4096 Dec  4 10:33 logs
drwx----- 2 nobody root 4096 Dec  3 19:33 proxy_temp
drwxr-xr-x 2 root  root 4096 Dec  3 19:12 sbin
drwx----- 2 nobody root 4096 Dec  3 19:33 scgi_temp
drwx----- 2 nobody root 4096 Dec  3 19:33 uwsgi_temp
```



接着来检查一下，输入地址 `http://192.168.64.10/testpage.html`



可以看到已经有页面显示，完成上传~

参考

[源代码安装Nginx \(nginx-1.22.1\)](#)

[Nginx 源码编译安装与运行](#)

[Ubuntu源码安装Nginx详解](#)

[如何在 Ubuntu 上安装和使用 Nginx](#)

Very interesting article: [How to Install Nginx from Source on Ubuntu](#)

[基于nginx上传一个简单的Html网页到服务器，并进行访问](#)

[How to Deploy a Simple Website with Nginx: A Comically Easy Guide](#)

[Install an NGINX web server on Ubuntu and create a website!](#)

Tomcat

Tomcat是一个开源的Web服务器和Servlet容器，它由Apache Software Foundation开发，主要用于运行Java Servlet和JavaServer Pages (JSP) 应用程序。Tomcat实现了Java EE规范中的Servlet和JSP部分，但并不包括EJB (Enterprise JavaBeans) 等其他部分，因此它通常被认为是一个轻量级的Web容器。

安装 Java

通过下面的命令安装 OpenJDK (默认的JDK)：

```
sudo apt install default-jdk
```

整个下载过程比较长，安装完毕后，使用 `java -version` 命令查看是否安装成功：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ java -version
openjdk version "21.0.5" 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mixed mode, sharing)
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

可以看到已经安装成功！

创建Tomcat用户

为了安全起见，您不应在没有唯一用户的情况下使用 Tomcat。这将使 Tomcat 在 Ubuntu 上的安装更加容易。创建一个将运行该服务的新 tomcat 组：

```
sudo groupadd tomcat
```

现在，下一步骤是创建一个tomcat用户。创建 Tomcat 组的用户成员，并使用主目录 opt/tomcat 来运行 Tomcat 服务：

```
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

- **禁用登录**：通过将 shell 设置为 `/bin/false`，该用户无法直接登录系统，只能用作服务账户运行 Tomcat。
- **组管理**：用户被添加到 `tomcat` 组，这样可以更好地管理与 Tomcat 相关的文件和权限。
- **隔离**：为 Tomcat 服务创建一个独立的账户，可以有效地限制该账户的权限，只允许它访问与 Tomcat 相关的文件和资源，从而提高系统的安全性。

如果想更加简洁的话，使用下面的命令：

```
sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat
```

- `-m`：自动创建该用户的主目录。系统会在默认的位置创建主目录，并将其作为该用户的家目录。在这里，它会在 `/opt/tomcat` 位置创建目录。
- `-U`：自动创建一个与用户同名的用户组。即，`tomcat` 用户会被添加到一个名为 `tomcat` 的组中。
- `-d`：指定用户的主目录。这里将 `tomcat` 用户的主目录设置为 `/opt/tomcat`，通常这是 Tomcat 安装目录所在的位置。

下载 Tomcat

接下来选择安装哪个Tomcat版本，这里我选择的是 [Tomcat 10版本](#)，命令如下：

```
sudo wget https://downloads.apache.org/tomcat/tomcat-10/v10.1.33/bin/apache-tomcat-10.1.33.tar.gz
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo useradd -m -U -d /opt/tomcat
-s /bin/false tomcat
[sudo] password for caicloudcat:
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo wget https://downloads.apache.org
/tomcat/tomcat-10/v10.1.33/bin/apache-tomcat-10.1.33.tar.gz -P /tmp
--2024-12-02 21:41:58-- https://downloads.apache.org/tomcat/tomcat-10/v10.1.33/bin/apach
e-tomcat-10.1.33.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.208.237, 135.181.214.104,
2a01:4f8:10a:39da::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.208.237|:443... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 13678161 (13M) [application/x-gzip]
Saving to: '/tmp/apache-tomcat-10.1.33.tar.gz'

apache-tomcat-10.1.33. 100%[=====>] 13.04M 27.7KB/s in 4m 30s

2024-12-02 21:46:29 (49.5 KB/s) - '/tmp/apache-tomcat-10.1.33.tar.gz' saved [13678161/136
78161]

caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

并将其解压到 `/opt/tomcat` 目录中：

```
sudo tar -xvf /tmp/apache-tomcat-10.1.33.tar.gz -C /opt/tomcat
```

说明：

- `-xvf`： `tar` 命令的三个选项
- `-x`：表示解压（extract）文件。告诉 `tar` 解压归档文件中的内容。
- `-v`：表示详细模式（verbose）。解压时，显示每个被解压的文件名。
- `-f`：表示文件（file），后面跟着归档文件的路径。`tar` 将会操作该文件。
- `-C` 选项告诉 `tar` 将文件解压到指定的目录。此处指定的是 `/opt/tomcat`，这意味着解压后的文件将被放置在 `/opt/tomcat` 目录下，如果你没有创建该目录，先手动创建它

更新Tomcat权限

将Tomcat目录的所有权更改为tomcat用户及组：

```
sudo chown -R tomcat:tomcat /opt/tomcat
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ cd /opt/  
caicloudcat@caicloudcat-VMware-Virtual-Platform:/opt$ ls -l  
total 4  
drwxr-x--- 3 tomcat tomcat 4096 Dec  2 21:50 tomcat  
caicloudcat@caicloudcat-VMware-Virtual-Platform:/opt$
```

配置Tomcat

在 `/etc/systemd/system` 目录下创建 systemd Unit file, 名为: `tomcat.service`。

执行下面的命令创建:

```
cd /etc/systemd/system  
sudo nano tomcat.service
```

将添加下面的内容:

```
[Unit]  
Description=Tomcat Server  
After=network.target  
  
[Service]  
Type=forking  
User=tomcat  
Group=tomcat  
Environment="JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64"  
WorkingDirectory=/opt/tomcat/apache-tomcat-10.1.33  
ExecStart=/opt/tomcat/apache-tomcat-10.1.24/bin/startup.sh  
  
[Install]  
WantedBy=multi-user.target
```

重新加载 systemd 并启动Tomcat

重新加载systemd守护进程以应用更改。

```
sudo systemctl daemon-reload
```

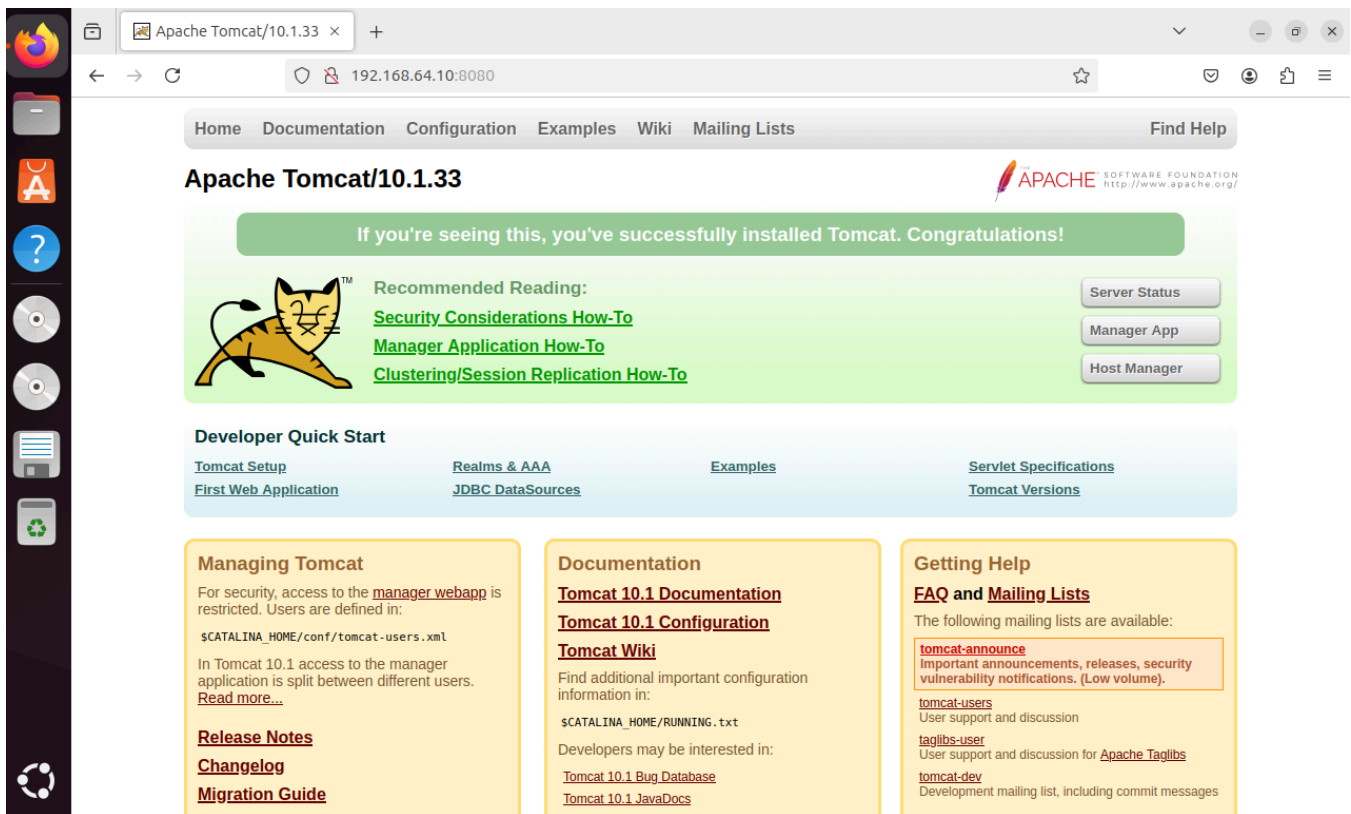
启动Tomcat服务：

```
sudo systemctl start tomcat
```

开启自启动服务：

```
sudo systemctl enable tomcat
```

接下来通过使用端口号8080的服务器IP地址来验证是否正常：



参考资料：

[How to Install Tomcat on Ubuntu 24.04 LTS](#)

[How to Install Tomcat on Ubuntu in 2024](#)

MySQL

介绍

MySQL的作用

MySQL是一款开源的关系型数据库管理系统（RDBMS），由瑞典MySQL AB公司开发，目前属于Oracle公司旗下产品。它以其体积小、速度快、成本低、开放源码等特点广受中小型网站和开发者的青睐，成为全球最受欢迎的数据库之一。

MySQL支持多种操作系统平台，包括Windows、Linux、macOS等，使得用户可以在不同环境下轻松部署和管理数据库。同时，MySQL还提供了简单易用的SQL语言接口，使得开发者可以方便地进行数据查询、插入、更新和删除等操作。

MySQL安装

方法一：通过apt-get安装MySQL

方法二：从源码包安装MySQL

在安装前，如果是使用虚拟机的话，推荐先试用快照保存安装前的状态，防止遇到难以出现的问题而造成不必要的麻烦。

在线安装MySQL

在线安装MySQL是比较方便的，只需要几个简单的命令即可安装MySQL服务，以 `Ubuntu 24.04 LTS` 版本来安装

使用apt-get包进行安装MySQL，详细可以参考其官方文档[Installing MySQL on Linux Using the MySQL APT Repository](#)

添加MySQL官方自带的Apt下载源

去官方下载源文件：

```
sudo wget -c -P /home https://repo.mysql.com//mysql-apt-config_0.8.32-1_all.deb
```

`wget` 是一个用于从网络上下载文件的命令行工具，支持多种网络协议，如 HTTP、HTTPS 和

FTP 等。

`-c` 参数表示支持断点续传。如果在下载过程中出现网络中断或其他问题导致下载中断，再次执行相同的 `wget` 命令加上 `-c` 参数，`wget` 会从上次中断的地方继续下载，而不是重新开始，这样可以节省时间和网络流量。

`-P /home`： `-P` 参数用于指定下载文件的保存目录。这里指定将文件下载到 `/home` 目录下。如果不指定该参数，`wget` 通常会将文件下载到当前工作目录。

在Ubuntu中安装刚下载的源：

会出现配置弹框，让你配置要安装啥MySQL版本，一般而言直接使用默认即可

```
sudo apt-get install /home/mysql-apt-config_0.8.32-1_all.deb
```

如果需要修改安装的MySQL版本（一般无需执行此命令，使用默认即可，但是想要其它MySQL版本方式则需要执行如下命令）

更改MySQL产品版本【会出现一个和上面一样的配置弹框】：

```
sudo dpkg-reconfigure mysql-apt-config
```

通过MySQL源来加载所选的包列表：

```
sudo apt-get update
```

使用Apt安装MySQL：

```
sudo apt-get install mysql-server
```

注：若安装是弹出“Enter root password:”则是密码输入框，输入两次即可，后面登录MySQL则使用此密码。

注：若安装是未弹出密码输入框，那么用户则需要通过无密码登录root用户，进去再修改密码。

到这里就已经成功安装了MySQL，通过下面的命令查看MySQL状态：

```
systemctl status mysql
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/home$ systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enable>
   Active: active (running) since Tue 2024-12-10 11:15:40 CST; 7min ago
   Process: 8537 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exit>
   Main PID: 8545 (mysqld)
   Status: "Server is operational"
   Tasks: 37 (limit: 4558)
   Memory: 364.3M (peak: 378.8M)
   CPU: 6.396s
   CGroup: /system.slice/mysql.service
           └─8545 /usr/sbin/mysqld

Dec 10 11:15:39 caicloudcat-VMware-Virtual-Platform systemd[1]: Starting mysql.ser>
Dec 10 11:15:40 caicloudcat-VMware-Virtual-Platform systemd[1]: Started mysql.serv>
lines 1-14/14 (END)
```

当然也可以使用 `mysql -V` 来检查是否安装成功

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ mysql -V
mysql Ver 8.0.40-0ubuntu0.24.04.1 for Linux on x86_64 ((Ubuntu))
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

离线方式安装MySQL

不是所有的服务器或者电脑都是有公网的，离线方式安装虽然难受，但也是必须要了解的；要将下载的包上传到服务器，在安装的同时遇到了缺少依赖问题，还得再去下载并上传，下面将介绍如何安装对应版本的MySQL。

Ubuntu和Debian缺少的依赖包可以在这个链接下载：pkgs.org/

首先去去下载 Ubuntu24.04 版本的MySQL，下载地址为：[MySQL下载](#)，进入后可以看到如下：

MySQL Enterprise Edition

MySQL Enterprise Edition includes the most comprehensive set of advanced features, management tools and technical support for MySQL.

[Learn More »](#)

[Customer Download from My Oracle Support \(MOS\) »](#) 客户端下载

[Trial Download from Oracle edelivery »](#) 试用下载

[Developer Download from Oracle OTN »](#)

MySQL NDB Cluster CGE

MySQL NDB Cluster is a real-time open source transactional database designed for fast, always-on access to data under high throughput conditions.

- MySQL NDB Cluster
- MySQL NDB Cluster Manager
- Plus, everything in MySQL Enterprise Edition

[Learn More »](#)

[Customer Download from My Oracle Support \(MOS\) »](#)

[Trial Download from Oracle edelivery »](#)

[MySQL Community \(GPL\) Downloads »](#) MySQL社区版下载，免费（我们下载这个）

我们进入MySQL社区版下载：

⬇ MySQL Community Downloads

- [MySQL Yum Repository](#) [关于yum方式安装下载](#)
- [MySQL APT Repository](#) [关于apt方式安装下载](#)
- [MySQL SUSE Repository](#)
- [MySQL Community Server](#) [MySQL社区版服务](#)
- [MySQL NDB Cluster](#) [MySQL集群版本](#)
- [MySQL Router](#)
- [MySQL Shell](#)
- [MySQL Operator](#)
- [MySQL NDB Operator](#)
- [MySQL Workbench](#) [图形界面管理工具](#)
- [MySQL Installer for Windows](#) [Windows下载](#)
- [C API \(libmysqlclient\)](#)
- [Connector/C++](#)
- [Connector/J](#)
- [Connector/NET](#)
- [Connector/Node.js](#)
- [Connector/ODBC](#)
- [Connector/Python](#)
- [MySQL Native Driver for PHP](#)
- [MySQL Benchmark Tool](#)
- [Time zone description tables](#)
- [Download Archives](#)

接着点击MySQL Community Server，进入详细下载：

MySQL Community Downloads

MySQL Community Server



MySQL Enterprise Edition for Developers

Free for learning, developing, and prototyping.

[Download Now »](#)

默认设置，最新发布的版本，不推荐

General Availability (GA) Releases

Archives



选择以前的MySQL版本

MySQL Community Server 9.1.0 Innovation

Select Version: 当前选择的版本

9.1.0 Innovation

Select Operating System: 选择安装的操作系统

Microsoft Windows

Windows (x86, 64-bit), MSI Installer

Windows的安装版，有界面的互动

9.1.0

118.1M

[Download](#)

(mysql-9.1.0-winx64.msi)

MD5: 7a26420bb3446eab56f389dba05a9718 | [Signature](#)

Windows (x86, 64-bit), ZIP Archive

解压版，需要手动配置

9.1.0

288.4M

[Download](#)

(mysql-9.1.0-winx64.zip)

MD5: 0a2333afc4ef07471bda89232697698e | [Signature](#)

Windows (x86, 64-bit), ZIP Archive

9.1.0

822.6M

[Download](#)

Debug Binaries & Test Suite

(mysql-9.1.0-winx64-debug-test.zip)

MD5: cb0327a504fc8d983f98c0cbf451a31d | [Signature](#)



We suggest that you use the [MD5 checksums and GnuPG signatures](#) to verify the integrity of the packages you download.

最后我们选择下载的面如下：

MySQL Community Downloads

MySQL Community Server



MySQL Enterprise Edition for Developers

Free for learning, developing, and prototyping.

[Download Now »](#)

General Availability (GA) Releases

Archives



MySQL Community Server 8.4.3 LTS

Select Version:

8.4.3 LTS

Select Operating System:

Ubuntu Linux

Select OS Version:

Ubuntu Linux 24.04 (x86, 64-bit)

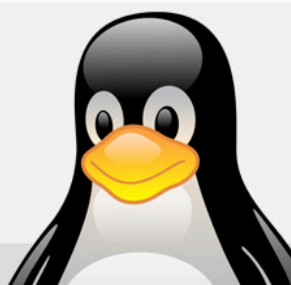
Install Using APT:

MySQL APT Repository

Supported Platforms:

- ▶ Debian
- ▶ Ubuntu

[Download Now »](#)



Download Packages:

DEB Bundle

8.4.3

474.7M

[Download](#)

(mysql-server_8.4.3-1ubuntu24.04_amd64.deb-bundle.tar)

MD5: 8e64ba9ce624fa9c0b08ad087422c | [Signature](#)

我们这里直接下载DEB Bundle的全量包，将然后解压进行进行安装。

将下载的MySQL8.4的deb包上传到服务器，我将其上传到 `/home/caicloudcat/Downloads` 目录下，然后通过dpkg安装。

```
tar -xvf mysql-server_8.4.3-1ubuntu24.04_amd64.deb-bundle.tar
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Downloads$ ls -l
total 486116
-rw-rw-r-- 1 caicloudcat caicloudcat 497776640 Dec 10 20:09 mysql-server_8.4.3-1ubuntu24.04_amd64.deb-bundle.tar
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Downloads$ tar -xvf mysql-server_8.4.3-1ubuntu24.04_amd64.deb-bundle.tar
libmysqlclient24_8.4.3-1ubuntu24.04_amd64.deb
libmysqlclient-dev_8.4.3-1ubuntu24.04_amd64.deb
mysql-client_8.4.3-1ubuntu24.04_amd64.deb
mysql-common_8.4.3-1ubuntu24.04_amd64.deb
mysql-community-client-plugins_8.4.3-1ubuntu24.04_amd64.deb
mysql-community-client-core_8.4.3-1ubuntu24.04_amd64.deb
mysql-community-test_8.4.3-1ubuntu24.04_amd64.deb
mysql-community-test-debug_8.4.3-1ubuntu24.04_amd64.deb
mysql-server_8.4.3-1ubuntu24.04_amd64.deb
mysql-community-server-core_8.4.3-1ubuntu24.04_amd64.deb
mysql-community-server-debug_8.4.3-1ubuntu24.04_amd64.deb
mysql-community-client_8.4.3-1ubuntu24.04_amd64.deb
mysql-testsuite_8.4.3-1ubuntu24.04_amd64.deb
mysql-community-server_8.4.3-1ubuntu24.04_amd64.deb
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Downloads$
```

`dpkg` 是一个非常重要的软件包管理工具，用于安装、卸载、查询和管理 `.deb` 软件包

```
dpkg [option] [package_name.deb]
```

其中，`option` 是各种可选操作参数，`package_name.deb` 是要操作的 `.deb` 软件包文件名或已安装软件包的名称。

常用参数及操作

- 安装软件包
 - 语法： `dpkg -i package.deb`
 - 示例： `dpkg -i myapp_1.0.0.deb`，将名为 `myapp_1.0.0.deb` 的软件包安装到系统中。
- 卸载软件包
 - 语法： `dpkg -r package_name` 或 `dpkg -P package_name`
 - 示例： `dpkg -r myapp`，将名为 `myapp` 的已安装软件包卸载，但保留其配置文件；若使用 `dpkg -P myapp`，则会彻底删除软件包及相关配置文件。
- 查询软件包信息
 - 查询已安装软件包的详细信息： `dpkg -s package_name`，例如 `dpkg -s firefox`，会显示 `firefox` 软件包的详细信息，包括版本、安装状态、依赖关系等。
 - 查询系统中所有已安装的软件包列表： `dpkg -l`，会列出系统中所有已安装软件包的名称、版本、架构等信息。
 - 查询指定软件包的安装位置： `dpkg -L package_name`，例如 `dpkg -L vim`，会显示 `vim` 软件包在系统中的安装目录及文件列表。

- 查询未安装软件包的信息： `dpkg -I package.deb` , 可以查看未安装的 .deb 软件包的详细信息, 如包名、版本、依赖关系等。
- 重新配置软件包
 - 语法： `dpkg-reconfigure package_name`
 - 示例： `dpkg-reconfigure locales` , 可以重新配置 locales 软件包的设置, 如语言环境等。

依赖关系处理

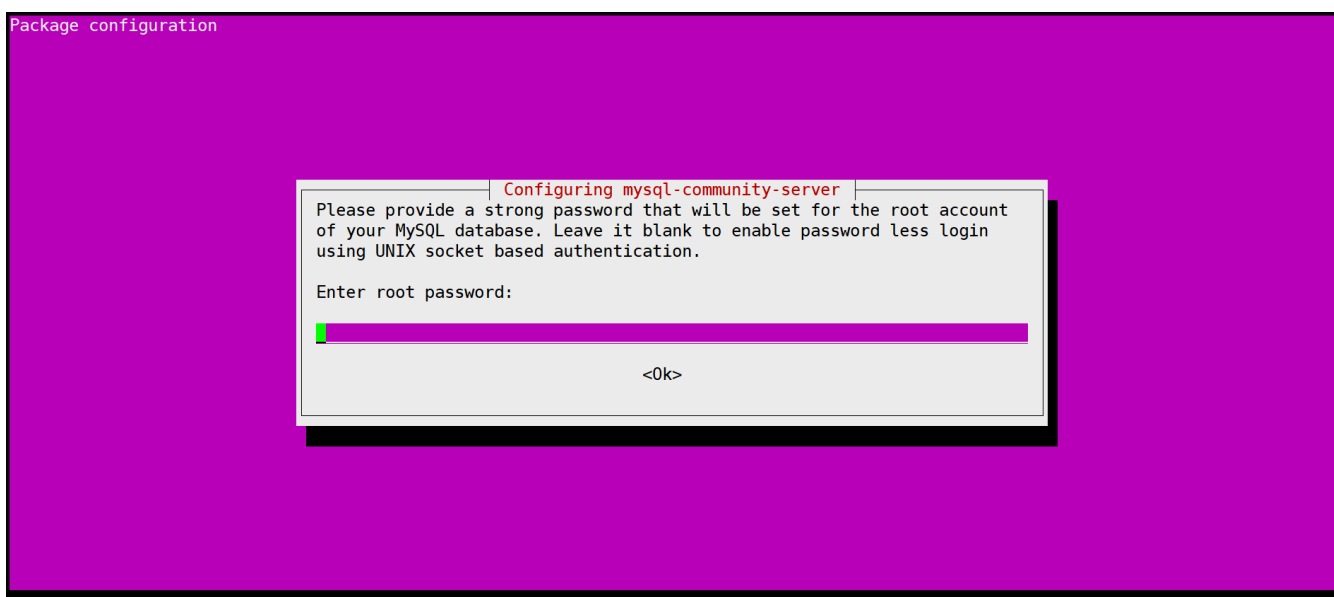
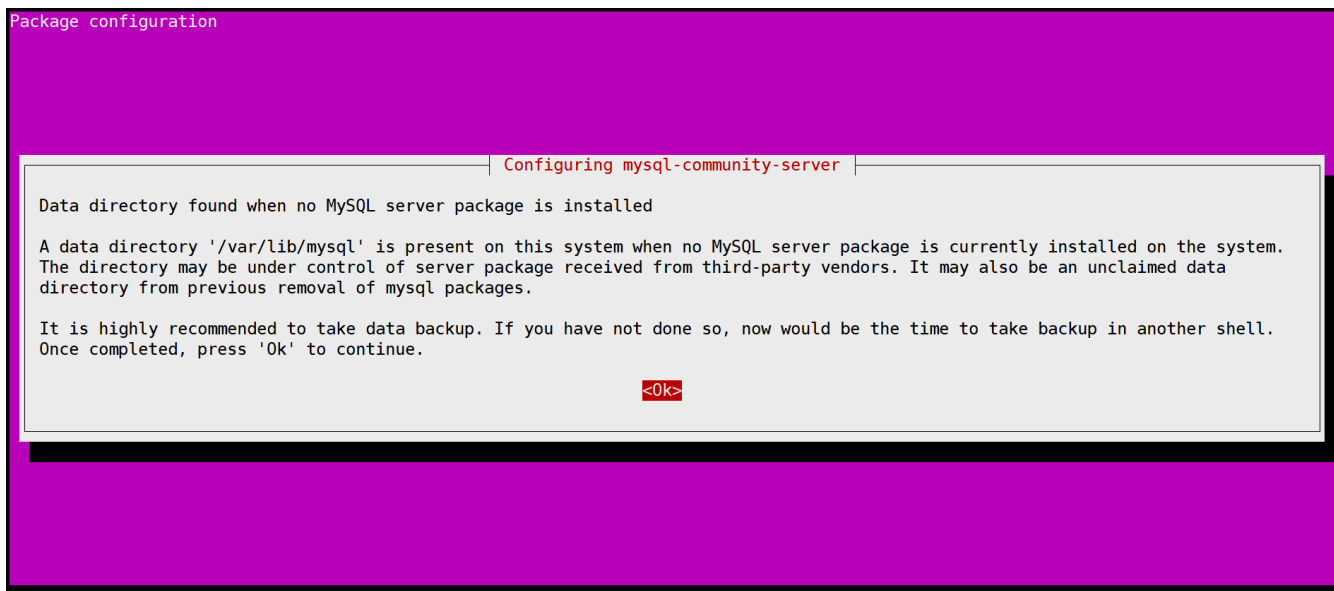
- `dpkg` 本身在处理软件包依赖关系时比较简单直接, 如果安装的软件包存在依赖问题, 可能会出现安装失败的情况。
- 通常在这种情况下, 需要先手动安装依赖的软件包, 或者使用更高级的包管理工具如 `apt` 来解决依赖问题, 因为 `apt` 在安装软件包时会自动处理依赖关系。

注意事项

- 使用 `sudo` 权限：大多数 `dpkg` 操作需要 `sudo` 权限, 因为这些操作涉及到系统软件包的安装和管理, 普通用户通常没有足够的权限进行这些操作。
- 谨慎卸载：在卸载软件包时要谨慎, 尤其是使用 `dpkg -P` 彻底删除软件包及配置文件时, 可能会导致一些配置丢失或系统功能异常, 特别是对于一些系统关键软件包。
- 依赖问题：安装软件包前最好先了解其依赖关系, 确保所需的依赖软件包已安装或可通过其他方式获取, 以避免安装失败。

使用 `dpkg` 的安装顺序：

```
sudo dpkg -i mysql-community-client-plugins_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i mysql-community-client-core_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i mysql-common_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i mysql-community-client_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i mysql-client_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i libmysqlclient24_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i libmysqlclient-dev_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i mysql-community-server-core_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i mysql-community-server_8.4.3-1ubuntu24.04_amd64.deb
sudo dpkg -i mysql-server_8.4.3-1ubuntu24.04_amd64.deb
# 如遇失败: sudo apt-get remove --purge ...
```



mysql server安装过程中会提示输入root用户密码，我这里使用的密码是 root ，待所有安装完成后，使用命令登陆即可。

通过 `mysql -V` 来检查是否安装成功：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Downloads$ mysql -V
mysql Ver 8.4.3 for Linux on x86_64 (MySQL Community Server - GPL)
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Downloads$
```

安装完成后，可以通过 `find / -name mysql` 来查看相关文件，下面是一些具体的主目录：

`/etc/mysql` 这是MySQL的主配置目录，存放MySQL服务器的配置文件（'my.cnf文件就在这'）

`/var/lib/mysql` 这是MySQL数据库的数据存储目录，包含实际的数据库文件（'重要的目录，我们创建的数据库数据都在这里'）

`/var/lib/mysql/mysql` 这是MySQL系统数据库mysql的专用目录

`/var/log/mysql` 这是MySQL的日志目录，存放MySQL服务器的日志文件

`/usr/lib/mysql` 这是MySQL库文件的存储目录

`/usr/bin/mysql` 这是MySQL客户端工具的可执行文件位置

`/usr/include/mysql` 这是MySQL开发库的头文件存放位置，用于开发与MySQL交互的应用程序

`/usr/include/mysql/mysql` 这个子目录通常包含MySQL库的更具体的头文件。

dpkg和apt-get有什么区别？

在 Ubuntu 及其他基于 Debian 的 Linux 系统中，`dpkg` 和 `apt-get` 都是常用的软件包管理工具，它们之间存在以下一些区别：

功能侧重

- **dpkg**：主要用于对本地的 `.deb` 软件包进行底层操作，如安装、卸载、查询等，侧重于对单个软件包文件的直接处理，不自动解决软件包之间的依赖关系。
- **apt-get**：是一个更高级的软件包管理工具，在 `dpkg` 的基础上进行了封装，不仅可以安装、卸载和查询软件包，还能自动处理软件包之间的依赖关系，从软件源中获取软件包并进行安装。

依赖关系处理

- **dpkg**：在安装或卸载软件包时，如果存在依赖关系问题，通常会直接报错并停止操作，需要用户手动解决依赖关系后再继续。
- **apt-get**：会自动检测并解决软件包的依赖关系，在安装一个软件包时，它会自动下载并安装该软件包所依赖的其他软件包，在卸载时也会根据依赖情况决定是否删除相关软件包。

软件源管理

- **dpkg**: 不直接涉及软件源的管理, 它只对本地的 `.deb` 文件进行操作, 安装软件包时需要用户手动指定本地软件包文件的路径。
- **apt-get**: 与软件源紧密相关, 通过配置的软件源来获取软件包列表和软件包本身。用户可以方便地使用 `apt-get` 命令添加、删除或更新软件源, 系统会根据软件源中的信息自动下载最新版本的软件包。

操作的便捷性与灵活性

- **dpkg**: 操作相对较为简单直接, 适合对单个 `.deb` 文件进行快速安装、卸载或查询等基本操作, 对于熟悉软件包内部结构和依赖关系的用户来说, 具有较高的灵活性。
- **apt-get**: 提供了更丰富的命令选项和功能, 使用起来更加方便和友好, 适合普通用户进行日常的软件安装、更新和卸载等操作。例如, `apt-get update` 用于更新软件源列表, `apt-get upgrade` 用于升级系统中所有可升级的软件包。

系统完整性和稳定性

- **dpkg**: 由于不自动处理依赖关系, 如果用户在使用 `dpkg` 安装或卸载软件包时不小心处理不当, 可能会导致系统中软件包的依赖关系混乱, 影响系统的稳定性和正常运行。
- **apt-get**: 在处理软件包操作时会更加谨慎和全面, 自动解决依赖关系有助于保持系统的完整性和稳定性, 降低因软件包管理不当而导致系统故障的风险。

卸载MySQL

Ubuntu卸载MySQL, 以 `Ubuntu24.04(LTS)` 为例

首先查看以及安装的MySQL信息

```
dpkg --get-selections | grep "mysql"
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:/var/lib$ dpkg --get-selections | grep "mysql"
mysql apt-config install
mysql-client-8.0 install
mysql-client-core-8.0 install
mysql-common install
mysql-server install
mysql-server-8.0 install
mysql-server-core-8.0 install
caicloudcat@caicloudcat-VMware-Virtual-Platform:/var/lib$
```

卸载查询出来的关于MySQL的依赖信息


```
sudo apt-get remove --purge mysql apt-config mysql-client-8.0 mysql-client-core-8.0 mysql-common m
```

卸载的过程中会弹出确认框，我们选择 Yes 即可卸载。

具体也可以参考这两篇文章：

[在 Ubuntu 中如何完全卸载 MySQL 服务器？](#)

[How to Uninstall MySQL from Ubuntu 24.04 \(for Beginners\)](#)

MySQL服务状态处理

注意：

- 在Ubuntu下的MySQL服务名为 `mysql.service`
- 在CentOS下的MySQL服务名为 `mysqld.service`

基本命令

```
systemctl status mysql.service 【查询MySQL在系统的状态】
systemctl start mysql.service 【启动MySQL服务】
systemctl stop mysql.service 【关闭MySQL服务】
systemctl restart mysql.service 【重启MySQL服务】
ps -ef | grep mysql 【查看MySQL进程】
```

其他命令

```
# 查询MySQL服务是否是自启动
systemctl list-unit-files | grep mysql.service
# enabled和disable分别为是和否

# 设置开机自启动
systemctl enable mysql.service
# 设置开机不自启动
systemctl disable mysql.service
```

也可以使用下面的命令开启或关闭服务

```
sudo service mysql start
sudo service mysql stop

sudo service mysql restart
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo service mysql stop
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo lsof -i:3306
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo service mysql start
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo lsof -i:3306
COMMAND PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
mysqld  3927 mysql  23u  IPv4  38956      0t0  TCP localhost:mysql (LISTEN)
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

远程连接MySQL

在本地（Windows系统）安装Navicat等客户端工具，并远程连接MySQL。

本机ping远程的IP地址

首先在本机ping一下远程的IP地址是否网络通畅

```
C:\Users\lenovo>ping 192.168.64.10
```

正在 Ping 192.168.64.10 具有 32 字节的数据：

来自 192.168.64.10 的回复：字节=32 时间<1ms TTL=64

来自 192.168.64.10 的回复：字节=32 时间=2ms TTL=64

来自 192.168.64.10 的回复：字节=32 时间<1ms TTL=64

来自 192.168.64.10 的回复：字节=32 时间<1ms TTL=64

192.168.64.10 的 Ping 统计信息：

数据包：已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，

往返行程的估计时间(以毫秒为单位)：

最短 = 0ms，最长 = 2ms，平均 = 0ms

防火墙放行

确保当前MySQL端口已被开放或者关闭了防火墙

Ubuntu的防火墙操作（ufw）：

```
# 查看防火墙状态
ufw status
# 开启或关闭防火墙
ufw enable
ufw disable
# 若开启了防火墙需要开启3306端口
ufw allow 3306 comment "MySQL端口放行"
# 更新防火墙规则
ufw reload
```

可以看到一开始防火墙是没有放行的：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo ufw status
[sudo] password for caicloudcat:
Status: active

To Action From
--
22/tcp ALLOW Anywhere
Nginx HTTP ALLOW Anywhere
Nginx HTTPS ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
Nginx HTTP (v6) ALLOW Anywhere (v6)
Nginx HTTPS (v6) ALLOW Anywhere (v6)
```

输入下面的命名后，即可打开：

```
sudo ufw allow 3306 comment "MySQL端口放行"
sudo ufw reload
```

```
Status: active

To Action From
--
22/tcp ALLOW Anywhere
Nginx HTTP ALLOW Anywhere
Nginx HTTPS ALLOW Anywhere
3306 ALLOW Anywhere # MySQL端口放行
22/tcp (v6) ALLOW Anywhere (v6)
Nginx HTTP (v6) ALLOW Anywhere (v6)
Nginx HTTPS (v6) ALLOW Anywhere (v6)
3306 (v6) ALLOW Anywhere (v6) # MySQL端口放行
```

MySQL命令行操作

进入MySQL

默认是没有密码的，但不能用当前用户进入，而是需要提权

注：需要提权，是因为没有启动服务，如果启动服务，后就不再需要提权了，下面也是。

```
sudo mysql -uroot -p
```

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ mysql -uroot -p
Enter password:
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$ sudo mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

如果是离线下载，这里会弹出错误：

```
caicloudcat@caicloudcat-VMware-Virtual-Platform:~/Downloads$ sudo mysql -uroot -p
Enter password:
ERROR 1524 (HY000): Plugin 'mysql_native_password' is not loaded
```

可以在 `etc/mysql/mysql.conf.d/mysqld.cnf` MySQL配置文件中添加 `mysql_native_password=ON` 再重启服务即可解决。

修改密码

在进入MySQL的命令行模式后，进行下面的操作：

```
alter user 'root'@'localhost' identified by 'root';

flush privileges;

exit;
```

```
mysql> alter user 'root'@'localhost' identified by 'root';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> exit;
Bye
caicloudcat@caicloudcat-VMware-Virtual-Platform:~$
```

再次使用 `sudo mysql -uroot -p` 命令，进入mysql，输入修改后的密码 `root`，即可进入mysql。

完成后，重启服务 `sudo service mysql restart`

先进入mysql，查看所有的数据库：

```
show database;
```

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+-----+
4 rows in set (0.01 sec)
```

选择mysql数据库：

```
use mysql;
```

允许任何IP远程连接：

```
update user set Host='%' where User='root';
```

在 MySQL 的用户表中，`Host` 字段表示允许用户登录的主机地址。

默认情况下，`root` 用户可能只允许从本地（`127.0.0.1`）登录。将 `Host` 字段的值设置为 `%`，这是一个通配符，表示允许 `root` 用户从任何 IP 地址登录，`Host=192.168.1.%` 表示只要是 IP 地址为 `192.168.1.*` 的客户端都可以连接。

通过 `update` 语句，可以修改用户表中的记录，更新 `root` 用户的主机访问权限。

设置密码规则：

```
ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY 'root';
```

在这里，将 `root` 用户（`'root'@'%'` 表示从任何主机访问的 `root` 用户）的认证方式设置为 `mysql_native_password`，并将密码设置为 `root`。

`mysql_native_password` 是 MySQL 的一种常见的密码认证方式，通过这种方式设置密码后，用户在登录时需要提供正确的密码才能访问数据库。

设置远程访问：

```
grant all privileges on *.* to 'root'@'%' with grant option;
```

`grant` 语句用于授予用户权限。`all privileges` 表示授予所有权限，`*.*` 表示对所有数据库和所有数据表（第一个 `*` 代表数据库，第二个 `*` 代表数据表）。`'root'@'%'` 表示从任何主机访问的 `root` 用户，`with grant option` 表示被授予权限的用户（这里是 `root` 用户）可以将这些权限再授予其他用户。这样就全面地设置了 `root` 用户从任何主机访问时的所有权限。

指令刷新并退出：

```
flush privileges;  
exit;
```

`flush privileges;` 命令用于刷新 MySQL 的权限缓存。当对用户权限进行修改（如通过 `update` 或 `grant` 语句）后，MySQL 不会立即生效这些更改，需要执行这个命令来使新的权限设置生效。

注释掉 `mysqld.cnf` 中 `bind-address=127.0.0.1`：

打开文件 `/etc/mysql/mysql.conf.d/mysqld.cnf`，设置成如下内容，即将其注释掉。

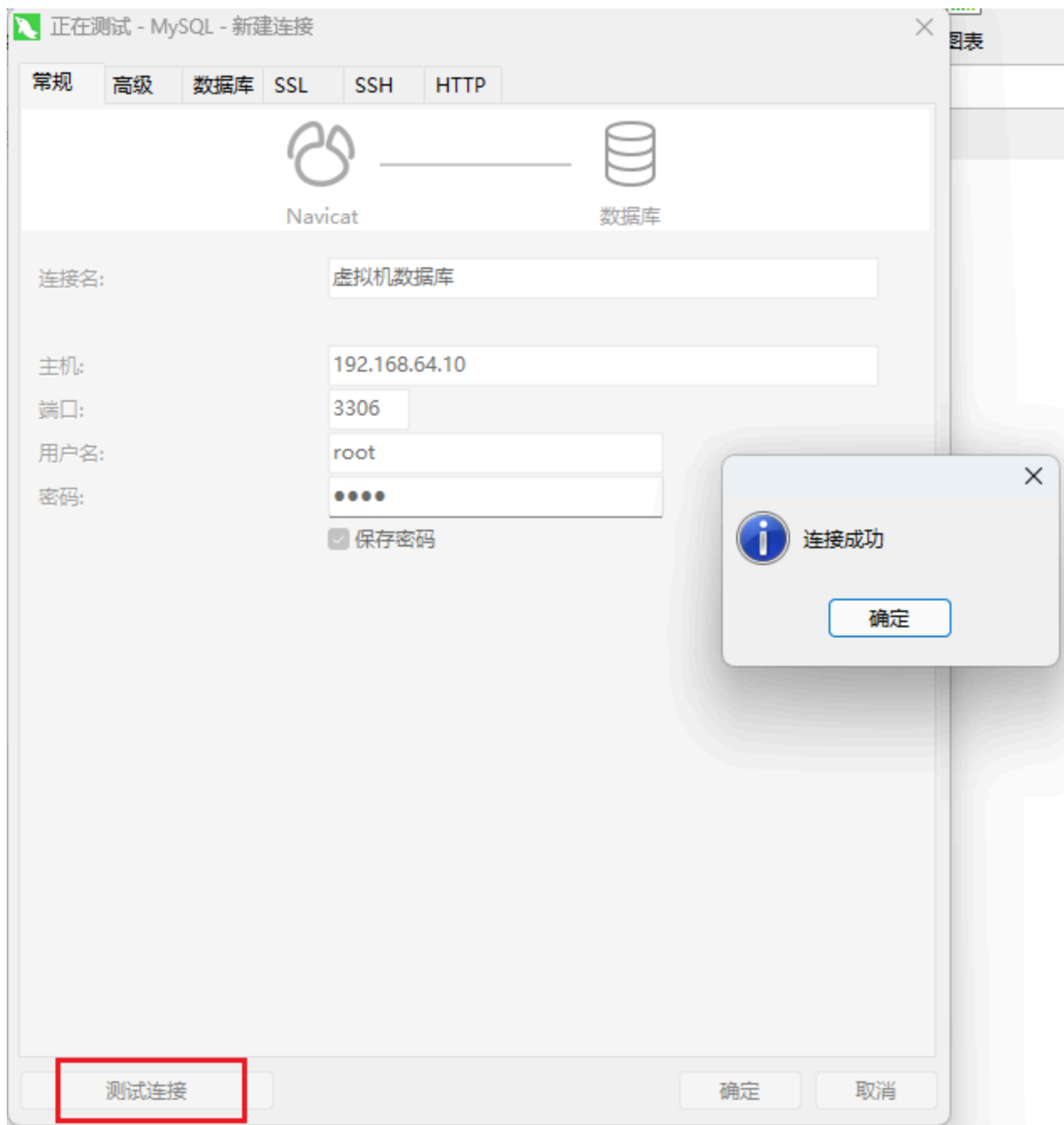
```
#bind-address=127.0.0.1  
mysqlx-bind-address = 127.0.0.1
```

在 MySQL 的配置文件 `etc/mysql/mysql.conf.d/mysqld.cnf` 中，`bind-address` 选项用于指定 MySQL 服务器监听的 IP 地址。默认情况下，设置为 `127.0.0.1`，这意味着 MySQL 服务器只接受来自本地（本机）的连接。通过将其注释掉，MySQL 服务器将监听所有可用的网络接口，从而允许来自其他主机（远程主机）的连接，前提是已经设置了用户权限允许远程访问。

最后重启服务：

```
sudo service mysql restart
```

在本机中的Navicat中测试连接，输入你的虚拟机IP以及密码，进行测试连接，若连接成功会有弹窗。



参考资料

[MySQL安装配置及卸载（超详细、各种安装方式）](#)

[ubuntu离线安装mysql](#)

[在 Ubuntu 中如何完全卸载 MySQL 服务器？](#)

[How to Uninstall MySQL from Ubuntu 24.04 \(for Beginners\)](#)