

Software Requirements

- Updated Ubuntu 22.04
- A user with sudo privileges

Hardware Requirements

- 4GB RAM
- 40GB Hard Disk

Pre-requisites

- Python 3.10
- Node.js 14+
- Redis 5 (caching and real time updates)
- MariaDB 10.3.x / Postgres 9.5.x (to run database driven apps)
- yarn 1.12+ (js dependency manager)
- pip 20+ (py dependency manager)
- wkhtmltopdf (version 0.12.5 with patched qt) (for pdf generation)
- NGINX (proxying multitenant sites in production)

First update and upgrade os

```
sudo apt update
```

```
sudo apt -y upgrade
```

Install Git

```
sudo apt install git
```

Install Python Tools & wkhtmltopdf

```
sudo apt-get install python3-dev
sudo apt-get install python3-setuptools python3-pip
sudo apt-get install xvfb libfontconfig wkhtmltopdf
sudo apt-get install libmysqlclient-dev
```

Install virtualenv

```
sudo apt-get install virtualenv
sudo apt install python3.10-venv
```

Install Curl, Redis and Node.js

```
sudo apt install curl
curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash
source ~/.profile
nvm install 14.15.0
```

Install Yarn

```
sudo apt-get install npm
sudo npm install -g yarn
```

install Redis

```
sudo apt-get install redis-server
```

Install MariaDB

```
sudo apt-get install software-properties-common  
sudo apt install mariadb-server  
sudo mysql_secure_installation
```

When you run this command, the server will show the following prompts. Please follow the steps as shown below to complete the setup correctly.

Enter current password for root: (Enter your SSH root user password)

Switch to unix_socket authentication [Y/n]: Y

Change the root password? [Y/n]: Y

It will ask you to set new MySQL root password at this step. This can be different from the SSH root user password.

Remove anonymous users? [Y/n] Y

Disallow root login remotely? [Y/n]: N

This is set as N because we might want to access the database from a remote server for using business analytics software like Metabase / PowerBI / Tableau, etc.

Remove test database and access to it? [Y/n]: Y

Reload privilege tables now? [Y/n]: Y

MySQL database development files

```
sudo apt-get install libmysqlclient-dev
```

Edit the mariadb configuration (unicode character encoding)

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Add this

```
[server]
user = mysql
pid-file = /run/mysqld/mysqld.pid
socket = /run/mysqld/mysqld.sock
basedir = /usr
datadir = /var/lib/mysql
tmpdir = /tmp
lc-messages-dir = /usr/share/mysql
bind-address = 127.0.0.1
query_cache_size = 16M
log_error = /var/log/mysql/error.log

[mysqld]
innodb-file-format=barracuda
innodb-file-per-table=1
innodb-large-prefix=1
character-set-client-handshake = FALSE
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
```

RESTART THE MYSQL

```
sudo service mysql restart
```

AGAIN RUN THIS COMMAND AND ADD THESE TEXT MENTIONED .

```
sudo nano /etc/mysql/my.cnf
```

ADD THIS

```
[mysqld]
```

```
character-set-client-handshake = FALSE
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
```

```
[mysql]
default-character-set = utf8mb4
```

Like this

```
# The MariaDB/MySQL tools read configuration files in the following order:
# 0. "/etc/mysql/my.cnf" symlinks to this file, reason why if you're using the .cnf extension
# 1. "/etc/mysql/mariadb.cnf" (this file) to set global defaults,
# 2. "/etc/mysql/conf.d/*.cnf" to set global options.
# 3. "/etc/mysql/mariadb.conf.d/*.cnf" to set MariaDB-only options.
# 4. "~/.my.cnf" to set user-specific options.
#
# If the same option is defined multiple times, the last one will apply.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# If you are new to MariaDB, check out
https://mariadb.com/kb/en/basic-mariadb-configuration-file/
#
# This group is read both by the client and the server
# use it for options that affect everything
#
#[client-server]
#
# Port or socket location where to connect
# port = 3306
# socket = /run/mysqld/mysqld.sock
#
# Import all .cnf files from configuration directory
#!includedir /etc/mysql/conf.d/
#!includedir /etc/mysql/mariadb.conf.d/
#
[mysqld]
character-set-client-handshake = FALSE
character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
```

```
# Add your additional MySQL server configurations below this line
# For example:
# innodb_buffer_pool_size = 2G
# max_connections = 200
# log_error = /var/log/mysql/error.log

[mysql]
default-character-set = utf8mb4
```

Save this and exit

Restart the my sql

```
sudo service mysql restart
```

Install frappe-bench

```
sudo -H pip3 install frappe-bench
bench --version
```

Initialize Frappe Bench

```
bench init --frappe-branch version-14 frappe-bench
```

Switch directories into the Frappe Bench directory

```
Cd frappe_bench
```

Then

Create new site

bench new-site [site-name]

Ex- bench new-site vibhanshu.com

Then use the site

bench use (site-name)

Ex- bench use vibhanshu.com

Set the bench configuration (adds a key value pair to site configuration file)

bench set-config developer_mode 1

Start your bench using bench command

Bench start

```
12:15:06 redis.cache.1 3735:M 22 Apr 2025 12:15:06.878 # Ready to accept connections
12:15:06 redis.queue.1 3740:C 22 Apr 2025 12:15:06.879 # 000000000000 Redis is starting 000000000000
12:15:06 redis.queue.1 3740:C 22 Apr 2025 12:15:06.879 # Redis version=6.0.16, bits=64, commit=00000000, modified=0, pid=3740, just started
12:15:06 redis.queue.1 3740:C 22 Apr 2025 12:15:06.879 # Configuration loaded
12:15:06 redis.queue.1 3740:M 22 Apr 2025 12:15:06.880 # Increased maximum number of open files to 10032 (it was originally set to 1024).
12:15:06 redis.queue.1 3740:M 22 Apr 2025 12:15:06.881 # Running mode=standalone, port=11000.
12:15:06 redis.queue.1 3740:M 22 Apr 2025 12:15:06.881 # Server initialized
12:15:06 redis.queue.1 3740:M 22 Apr 2025 12:15:06.881 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix
x 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.
12:15:06 redis.queue.1 3740:M 22 Apr 2025 12:15:06.881 # Ready to accept connections

12:15:07 watch.1 yarn run v1.22.22
12:15:07 watch.1 $ node rebuild watch --live-reload
12:15:07 watch.1 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
12:15:08 web.1 * Running on all addresses (0.0.0.0)
12:15:08 web.1 * Running on http://127.0.0.1:8000
12:15:08 web.1 * Running on http://172.31.44.47:8000
12:15:08 web.1 Press CTRL+C to quit
12:15:08 web.1 * Restarting with stat
12:15:08 web.1 * Debugger is active!
12:15:08 web.1 * Debugger PIN: 234-998-483
12:15:09 watch.1 clean: postcss.plugin was deprecated. Migration guide:
12:15:09 watch.1 https://evilmartians.com/chronicles/postcss-8-plugin-migration
12:15:12 watch.1 Browserslist: caniuse-lite is outdated. Please run:
12:15:12 watch.1 npx browserslist@latest --update-db
12:15:12 watch.1 Why you should do it regularly: https://github.com/browserslist/browserslist#browsers-data-updating
12:15:21 watch.1 clean: postcss.plugin was deprecated. Migration guide:
12:15:21 watch.1 https://evilmartians.com/chronicles/postcss-8-plugin-migration
12:15:23 watch.1 clean: postcss.plugin was deprecated. Migration guide:
12:15:23 watch.1 https://evilmartians.com/chronicles/postcss-8-plugin-migration
12:15:24 watch.1 clean: postcss.plugin was deprecated. Migration guide:
12:15:24 watch.1 https://evilmartians.com/chronicles/postcss-8-plugin-migration
12:15:24 watch.1 Watching for changes...
12:16:00 schedule.1 /home/ubuntu/frappe-bench/env/Lib/python3.10/site-packages/rq/connections.py:187: DeprecationWarning: The 'resolve_connection' func
tion is deprecated. Pass the 'connection' explicitly instead.
```