**Frappe/ERPNext Installation Guide for Ubuntu 22.04 LTS**

**Overview**

This comprehensive guide will walk you through the installation of Frappe Framework and ERPNext version 13 on Ubuntu 22.04 LTS. The installation includes all necessary dependencies and configurations for a fully functional development environment.

**System Requirements**

**Hardware Requirements**

* **RAM**: Minimum 4GB (8GB recommended)
* **Storage**: Minimum 20GB free space
* **CPU**: 2+ cores recommended

**Software Requirements**

|  |  |  |
| --- | --- | --- |
| **Tool** | **Version** | **Purpose** |
| Python | 3.6+ (Ubuntu 22.04 uses 3.10.x) | Backend language |
| Node.js | 14.15.0 | Frontend build tools |
| Redis | 5+ | Caching, real-time updates |
| MariaDB | 10.3.x | Database |
| Yarn | 1.12+ | JS dependency manager |
| Pip | 20+ | Python package manager |
| wkhtmltopdf | 0.12.5 (qt-patched version) | PDF generation |
| NGINX + Cron | Optional | Reverse proxy and scheduled jobs |

**Installation Steps**

**Step 1: Update System and Install Git**

**sudo apt update && sudo apt upgrade -y**

**sudo apt install git -y**

**Step 2: Install Python Development Tools**

**sudo apt install pkg-config python3-dev python3-pip python3-setuptools -y**

**Step 3: Install Virtual Environment**

**sudo apt install virtualenv -y**

**Step 4: Verify Python Version and Install venv**

**python3 -V**

If Python 3.10 is installed, run:

**sudo apt install python3.10-venv -y**

**Step 5: Install and Configure MariaDB**

**Install MariaDB**

**sudo apt install software-properties-common mariadb-server -y**

**Secure MariaDB Installation**

**sudo mysql\_secure\_installation**

Follow the prompts:

* Enter current password for root: **(Leave blank or enter SSH root password)**
* Switch to unix\_socket authentication [Y/n]: **Y**
* Change the root password? [Y/n]: **Y (set a new password)**
* Remove anonymous users? [Y/n]: **Y**
* Disallow root login remotely? [Y/n]: **N (set to N if remote access is needed)**
* Remove test database and access to it? [Y/n]: **Y**
* Reload privilege tables now? [Y/n]: **Y**

**Configure MariaDB for Frappe**

Edit the MariaDB configuration file:

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

Add the following configuration:

**[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**

**[mysql]**

**default-character-set = utf8mb4**

Restart MariaDB service:

**sudo service mysql restart**

**Step 6: Install Redis**

**sudo apt install redis-server -y**

Verify Redis is running:

**redis-cli ping**

Expected output: **PONG**

**Step 7: Install Node.js 14.x using NVM**

**Install NVM (Node Version Manager)**

**sudo apt install curl -y**

**~~#curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash~~**

**curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash**

**source ~/.bashrc**

**Verify nvm installation:**

**nvm --version**

If nvm command is not found, run:

**source ~/.nvm/nvm.sh**

**Install Node.js 18.xx.x (18.20.8)**

**nvm install 18**

**nvm use 18**

**nvm alias default 18**

Verify installation:

**node -v**

**npm -v**

**# 1. Update package index**

**sudo apt update**

**# 2. Install curl (if not already installed)**

**sudo apt install curl -y**

**# 3. Download and run the NodeSource setup script for Node.js 18 (recommended LTS)**

**curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E bash -**

**# 4. Install Node.js and npm**

**sudo apt install -y nodejs**

**# 5. Confirm installation**

**node -v**

**npm -v**

**sudo npm install -g yarn**

**Step 8: Install Yarn**

**sudo npm install -g yarn**

Verify installation:

**yarn --version**

**Step 9: Install wkhtmltopdf**

**sudo apt install xvfb libfontconfig wkhtmltopdf -y**

**Note**: For production environments, it's recommended to use wkhtmltopdf 0.12.5 with qt patch for better PDF rendering.

**Step 10: Install MySQL/MariaDB Development Libraries**

**sudo apt install libmysqlclient-dev default-libmysqlclient-dev**

**sudo apt update**

**sudo apt install libmariadb-dev libmariadb-dev-compat**

**(If using MariaDB instead of MySQL, also run:** **sudo apt install libmariadb-dev)**

**Step 11: Install Frappe Bench**

**sudo -H pip3 install frappe-bench**

Verify installation:

**bench --version**

**Step 12: Initialize Frappe Bench**

**bench init --frappe-branch version-14 frappe-bench**

**Note: frappe and expnext version must be same**

**~~bench init frappe-bench --frappe-branch version-13~~ --Required Version**

**cd frappe-bench**

**Step 13: Create a New Site**

**bench new-site your-site-name.local**

if get any errors like **MySQLdb.OperationalError: (1698, "Access denied for user 'root'@'localhost'")**

**sudo mysql**

**SET PASSWORD FOR 'root'@'localhost' = PASSWORD('root');**

**FLUSH PRIVILEGES;**

**MariaDB [(none)]> SELECT user, host, plugin FROM mysql.user WHERE user = 'root';**

**+------+-----------+-----------------------+**

**| User | Host | plugin |**

**+------+-----------+-----------------------+**

**| root | localhost | mysql\_native\_password |**

**+------+-----------+-----------------------+**

**1 row in set (0.113 sec)**

**MariaDB [(none)]> ALTER USER 'root'@'localhost' IDENTIFIED BY 'root';**

**MariaDB [(none)]> FLUSH PRIVILEGES;**

**MariaDB [(none)]> EXIT;**

**Bye**

**mysql -u root -proot -e "SELECT 1"**

**+---+**

**| 1 |**

**+---+**

**| 1 |**

**+---+**

**Ex:- bench new-site gopi.local**

**Note: Replace your-site-name.local with your desired site name.**

You'll be prompted to enter:

* MySQL root password
* Administrator password for the site

**Step 14: Set Default Site**

**bench use your-site-name.local**

**npm install @tiptap/core@^2.0.0 –save**

**Step 15: Install ERPNext Application**

**Download ERPNext and hrms**

**bench get-app erpnext --branch version-14**

**bench get-app hrms**

**Install ERPNext and hrms on Site**

**bench --site your-site-name.local install-app erpnext**

**bench --site your-site-name.local install-app hrms**

**Step 16: Start Development Server**

**bench start**

Your ERPNext installation will be available at: http://localhost:8000

**Post-Installation Configuration**

**Creating a User (Optional)**

sudo adduser frappe

sudo usermod -aG sudo frappe

su - frappe

**Production Setup (Optional)**

For production deployment, run:

sudo bench setup production frappe

This will:

* Configure NGINX as reverse proxy
* Set up systemd services
* Configure SSL (optional)
* Set up automatic startup

**Troubleshooting**

**Common Issues and Solutions**

1. **Permission Errors**
2. sudo chown -R $USER:$USER /path/to/frappe-bench
3. **Port Already in Use**
4. bench set-config -g default\_site your-site-name.local
5. bench set-config -g serve\_default\_site 1
6. **MariaDB Connection Issues**
   * Ensure MariaDB is running: sudo systemctl status mariadb
   * Check configuration in /etc/mysql/mariadb.conf.d/50-server.cnf
7. **Redis Connection Issues**
   * Ensure Redis is running: sudo systemctl status redis
   * Check Redis configuration: redis-cli ping

**Useful Commands**

**Bench Commands**

**# Start development server**

**bench start**

**# Stop all services**

**bench stop**

**# Restart services**

**bench restart**

**# Update all apps**

**bench update**

**# Backup site**

**bench --site your-site-name.local backup**

**# Restore backup**

**bench --site your-site-name.local restore /path/to/backup**

**# Create new app**

**bench new-app app-name**

**# Install app to site**

**bench --site your-site-name.local install-app app-name**

**# List sites**

**bench --site all list-apps**

**# Enable/Disable maintenance mode**

**bench --site your-site-name.local set-maintenance-mode on**

**bench --site your-site-name.local set-maintenance-mode off**

**Database Commands**

# Access MariaDB console

bench --site your-site-name.local mariadb

# Database console

bench --site your-site-name.local db-console

**Security Considerations**

**For Production Environments**

1. **Change default passwords**
2. **Enable firewall**
3. sudo ufw enable
4. sudo ufw allow ssh
5. sudo ufw allow http
6. sudo ufw allow https
7. **Regular backups**
8. bench --site your-site-name.local backup --with-files
9. **SSL Configuration**
10. sudo bench setup lets-encrypt your-site-name.local

**Maintenance**

**Regular Updates**

# Update bench

pip3 install --upgrade frappe-bench

# Update all apps

bench update

# Update specific app

bench update --app erpnext

**Log Files**

Important log locations:

* Frappe logs: ~/frappe-bench/logs/
* MariaDB logs: /var/log/mysql/error.log
* NGINX logs: /var/log/nginx/

**Conclusion**

You now have a fully functional Frappe/ERPNext installation on Ubuntu 22.04 LTS. The system is ready for development and can be configured for production use with additional security measures.

For further customization and advanced configurations, refer to the official Frappe documentation at <https://frappeframework.com/docs>.

**Support**

If you encounter issues:

1. Check the troubleshooting section above
2. Review log files for error messages
3. Consult the official documentation
4. Visit the community forum at <https://discuss.frappe.io/>

**Document Version**: 1.0  
**Last Updated**: June 2025  
**Compatible with**: Ubuntu 22.04 LTS, Frappe v13, ERPNext v13

**Reinstall Site (Recommended for New Setups)**

bench drop-site gopi.local --root-password your\_mysql\_root\_password

bench new-site gopi.local

bench --site gopi.local install-app erpnext

bench --site gopi.local backup

bench --site gopi.local install-app erpnext

bench --site gopi.local install-app hrms

bench drop-site deepgrid.local

bench drop-site deepgrid.local --force

bench --site deepgrid.local reinstall

bench --site deepgrid.local uninstall-app hrms

bench --site deepgrid.local uninstall-app hrms --force

bench --site deepgrid.local uninstall-app hrms --delete

bench remove-app hrms

rm -rf apps/hrms

# Step 1: Create a new bench

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

cd frappe-bench

# Step 2: Create a site

bench new-site yoursite.local

# Step 3: Get compatible ERPNext and HRMS apps

bench get-app --branch version-14 erpnext https://github.com/frappe/erpnext

bench get-app --branch version-14 hrms https://github.com/frappe/hrms

# Step 4: Install apps on the site

bench --site yoursite.local install-app erpnext

bench --site yoursite.local install-app hrms

# Step 5: Start the server

bench start

**# Step 1: Create a new bench**

**bench init frappe-bench --frappe-branch version-14**

**cd frappe-bench**

**# Step 2: Create a site**

**bench new-site yoursite.local**

**# Step 3: Get compatible ERPNext and HRMS apps**

**bench get-app --branch version-14 erpnext https://github.com/frappe/erpnext**

**bench get-app --branch version-14 hrms https://github.com/frappe/hrms**

**# Step 4: Install apps on the site**

**bench --site yoursite.local install-app erpnext**

**bench --site yoursite.local install-app hrms**

**# Step 5: Start the server**

**bench start**

You're **almost there**, but Redis is still failing with spawn error → BACKOFF. This means something is still crashing at Redis startup.

Now that you've **regenerated the Redis configs**, let’s debug it **cleanly**:

**✅ Step-by-Step Debug for Redis spawn error**

**🔍 Step 1 — Check port availability**

Make sure the Redis ports are **not** in use:

bash

CopyEdit

sudo lsof -i :11002 -i :13002 -i :13001

If **any** of them show a process, kill it:

sudo kill -9 <PID>

Then run again until you get **no output**.

**🧹 Step 2 — Clean up stale Redis PID files**

rm -f ~/frappe-bench-v15/config/redis\_\*.pid

rm -f /tmp/redis-\*.pid

**📜 Step 3 — Manually run a Redis server (test config)**

Let's manually test the queue Redis:

redis-server config/redis\_queue.conf

If this shows a fatal error like:

Could not create server TCP listening socket 127.0.0.1:11002: bind: Address already in use

→ then the port is still occupied.

If it starts successfully:

* Press Ctrl+C to stop it
* That means Supervisor **should** also be able to start it

**📂 Step 4 — Tail Redis logs (this is key)**

Run this and **share the last few lines**:

tail -n 30 logs/redis-queue.log

tail -n 30 logs/redis-cache.log

This will tell us **exactly why** it's failing (e.g. port issue, file permission, config error).

**🟢 Step 5 — Start Redis via Supervisor again**

After the above steps:

sudo supervisorctl start "frappe-bench-v15-redis:\*"

Then confirm:

sudo supervisorctl status | grep redis

**📌 Summary**

If you paste the output of the following, I can tell you the root cause immediately:

sudo lsof -i :11002 -i :13002 -i :13001

tail -n 30 logs/redis-queue.log

tail -n 30 logs/redis-cache.log

Once Redis is fixed, the rest of the system (Node-SocketIO, attendance API, Axios) will fall into place smoothly.

**bench setup redis # regenerates redis conf files from**

**common\_site\_config.json**

**bench setup procfile # updates Procfile with correct ports**