

# Odoo

- [Odoo18](#)
- [odoo18 ubuntu 24.02](#)

# Odoo18

```
sudo apt-get update
```

```
sudo apt-get upgrade -y
```

```
sudo apt install software-properties-common
```

```
sudo add-apt-repository ppa:deadsnakes/ppa
```

```
sudo apt update
```

## Creating the user

```
sudo useradd -m -d /opt/odoo18 -U -r -s /bin/bash odoo18
```

```
apt list | grep python3.12
```

```
sudo apt-get install python3.12
```

```
sudo apt install -y git python3-pip python3.12-dev libxml2-dev libxslt1-dev zlib1g-dev libsasl2-dev libldap2-dev  
build-essential libssl-dev libffi-dev libmysqlclient-dev libjpeg-dev libpq-dev libjpeg8-dev liblcms2-dev libblas-dev  
libatlas-base-dev
```

## creating the web dependency

```
sudo apt-get install -y npm  
sudo ln -s /usr/bin/nodejs/usr/bin/node  
sudo npm install -g less less-plugin-clean-css  
sudo apt-get install -y node-less
```

## html to pdf

```
sudo wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.5/wkhtmltox_0.12.5-  
1.bionic_amd64.deb
```

```
sudo dpkg -i wkhtmltox_0.12.5-1.bionic_amd64.deb
```

```
sudo apt install -f
```

## Install and configure PostgreSQL

```
sudo apt install postgresql -y
```

```
sudo su - postgres
```

```
createuser --createdb --username postgres --no-createrole --no-superuser --pwprompt odoo18
```

```
psql  
ALTER USER odoo18 WITH SUPERUSER;
```

```
\q  
exit
```

## Install Odoo

```
sudo su - odoo18
```

```
git clone https://www.github.com/odoo/odoo --depth 1 --branch 18.0 odoo18
```

```
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.12 1
```

```
python3 -m venv odoo18-venv
```

```
source odoo18-venv/bin/activate
```

```
(odoo18-venv) odoo18@ubuntu22:~$ pip3 install wheel
```

```
(odoo18-venv) odoo18@ubuntu22:~$ pip3 install -r odoo18/requirements.txt
```

## Create a directory for the 3rd party addons:

```
mkdir /opt/odoo18/odoo18/custom-addons
```

## Create a configuration file for the Odoo Installation

```
sudo nano /etc/odoo18.conf
```

```
[options]
admin_passwd = admin_passwd
db_host = False
db_port = False
db_user = odoo18
db_password = False
addons_path = /opt/odoo18/odoo18/addons,/opt/odoo18/odoo18/custom-addons
xmlrpc_port = 8069
```

## Create Odoo Systemd Unit file

```
sudo nano /etc/systemd/system/odoo18.service
```

```
[Unit]
Description=Odoo18
Requires=postgresql.service
```

```
After=network.target postgresql.service
```

```
[Service]
```

```
Type=simple
```

```
SyslogIdentifier=odoo18
```

```
PermissionsStartOnly=true
```

```
User=odoo18
```

```
Group=odoo18
```

```
ExecStart=/opt/odoo18/odoo18-venv/bin/python3 /opt/odoo18/odoo18/odoo-bin -c /etc/odoo18.conf
```

```
StandardOutput=journal+console
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable --now odoo18
```

```
sudo systemctl status odoo18
```

```
sudo journalctl -u odoo18
```

# odoo18 ubuntu 24.02

## Step 3: Secure the Server

Securing your server is crucial to protect it from unauthorized access and potential attacks. Follow these steps to enhance your server's security:

**Install the OpenSSH Server:** This package ensures that your server can accept SSH connections, which is essential for remote management.

```
sudo apt-get install openssh-server
```

**Install Fail2Ban:** This tool helps protect your server from brute-force attacks by monitoring log files for failed login attempts and banning suspicious IP addresses.

```
sudo apt-get install fail2ban
```

After installation, start the Fail2Ban service and enable it to start automatically on boot.

```
sudo systemctl start fail2ban  
sudo systemctl enable fail2ban
```

**Verify Fail2Ban Status:** To ensure that Fail2Ban is running correctly, you can check its status with:

```
sudo systemctl status fail2ban
```

Securing your server with these steps will help protect it from common threats and unauthorized access. Once your server is secured, you can proceed with the Odoo 18 installation.

## Step 4: Install Packages and Libraries

Install several essential packages and libraries to ensure that Odoo 18 functions correctly. Follow these steps:

**Install Python 3 Pip:** Pip is the package installer for Python, and it's required to manage Python libraries.

```
sudo apt-get install -y python3-pip
```

**Install Development Libraries and Dependencies:** These libraries are necessary for building and running Odoo and its dependencies.

```
sudo apt-get install -y python3-dev libxml2-dev libxslt1-dev zlib1g-dev libsasl2-dev libldap2-dev build-essential libssl-dev libffi-dev libmysqlclient-dev libjpeg-dev libpq-dev libjpeg8-dev libcms2-dev libblas-dev libatlas-base-dev
```

**Install Node.js and NPM:** Node.js is required for various frontend tasks, and NPM is its package manager.

```
sudo apt-get install -y npm
```

**Create a Symlink for Node.js:** Sometimes, Node.js is installed as nodejs but some applications expect node. Create a symlink to ensure compatibility.

```
sudo ln -s /usr/bin/nodejs /usr/bin/node
```

if it already exist then check the version

```
node -v
```

**Install Less and Less Plugin for Clean CSS:** Less is a CSS pre-processor, and the clean CSS plugin helps minify CSS files.

```
sudo npm install -g less less-plugin-clean-css
```

**Install Node-Less:** This package integrates Less with Node.js.

```
sudo apt-get install -y node-less
```

These steps install all necessary packages and libraries to support the Odoo 18 installation. Once completed, you'll have all the prerequisites in place for the next steps in the installation process.

# Step 5: Set Up the Database Server

Odoo 18 requires PostgreSQL as its database management system. Follow these steps to install and configure PostgreSQL:

Install PostgreSQL: This command installs the PostgreSQL database server.

```
sudo apt-get install -y postgresql
```

```
sudo su - postgres
```

Create a new Database User: Create a user for Odoo with permission to create databases. You'll be prompted to enter a password for this user.

```
createuser --createdb --username postgres --no-createrole --superuser --pwprompt odoo18
```

-createdb: Allows the user to create databases.

--username postgres: Specifies the PostgreSQL superuser.

--no-createrole: Prevents the user from creating roles.

--superuser: Grants superuser privileges.

--pwprompt: Prompts for a password for the new user.

odoo18: The name of the new user.

Exit the PostgreSQL User Session: Return to your regular user account.

```
exit
```

# Step 6: Create a System User for Odoo

A dedicated system user for Odoo ensures that the application runs with the appropriate permissions and does not interfere with other system processes.

**Add a New System User:** Create a system user with its home directory where Odoo will be installed.

```
sudo adduser --system --home=/opt/odoo18 --group odoo18
```

`--system`: Creates a system user with a lower UID.

`--home=/opt/odoo18`: Specifies the home directory for the user.

`--group odoo18`: Creates a group with the same name and assigns the user to it.

This setup ensures that Odoo has a dedicated user and database for managing its operations securely. You're now ready to proceed with the Odoo installation and configuration.

## Step 7: Get Odoo 18 Community Edition from GitHub

To install Odoo 18, you need to clone the Odoo repository from GitHub. Follow these steps:

**Install Git:** Git is a version control system that you'll need to clone the Odoo repository.

```
sudo apt-get install -y git
```

**Switch to the Odoo System User:** Log in as the Odoo system user that you created in Step 6. This ensures that the Odoo files are owned by the correct user.

```
sudo su - odoo18 -s /bin/bash
```

**Clone the Odoo Repository:** Use Git to clone the Odoo 18 community edition repository. The `--depth 1` option ensures that you only clone the latest commit, and `--branch master` ensures that you get the latest stable release of the Odoo 18 branch.

```
git clone https://www.github.com/odoo/odoo --depth 1 --branch 18.0 --single-branch .
```

**Exit the Odoo User Session:** Return to your regular user account.

```
exit
```

Now that you have cloned the Odoo repository, you're ready to proceed with the configuration and installation of Odoo 18.

# Step 8: Install Required Python Packages

To ensure that Odoo 18 runs smoothly, you need to set up a Python virtual environment and install the required packages, as well as additional dependencies. Follow these steps:

1. Install Python 3 Virtual Environment Package: This package allows you to create isolated Python environment.

```
sudo apt install -y python3-venv
```

2. Create a Python Virtual Environment: Set up a virtual environment in the `/opt/odoo18/` directory to manage dependencies separately from the system Python.

```
sudo python3 -m venv /opt/odoo18/venv
```

3. Activate the Virtual Environment: Switch to the Odoo system user and activate the virtual environment.

```
sudo -s
```

```
cd /opt/odoo18/
```

```
source venv/bin/activate
```

4. Install Python Dependencies: Use pip to install the required Python packages listed in the `requirements.txt` file.

```
pip install -r requirements.txt
```

5. Install wkhtmltopdf: Odoo requires `wkhtmltopdf` to generate PDF reports. Download and install the `.deb` package for this utility.

```
sudo wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.5/wkhtmltox_0.12.5-1.bionic_amd64.deb
```

6. Install OpenSSL Dependency: Download and install the OpenSSL library if required.

```
sudo wget http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1f-1ubuntu2_amd64.deb
```

```
sudo dpkg -i libssl1.1_1.1.1f-1ubuntu2_amd64.deb
```

7. Install Additional Fonts: Install font packages needed by `wkhtmltopdf`.

```
sudo apt-get install -y xfonts-75dpi
```

8. Install wkhtmltopdf Package: Install the `wkhtmltopdf` package you downloaded.

```
sudo dpkg -i wkhtmltox_0.12.5-1.bionic_amd64.deb
```

9. Fix Dependency Issues: If there are any missing dependencies, this command will resolve them.

```
sudo apt install -f
```

10. Deactivate the Virtual Environment: Once you've completed the setup, deactivate the virtual environment.

```
deactivate
```

These steps ensure that all necessary Python packages and system dependencies are correctly installed, setting up your environment for the Odoo 18 installation.

## Step 9: Set Up the Configuration File

To configure Odoo 18, you need to create and edit the configuration file. This file contains important settings for the Odoo server, such as database connection details and logging options.

1. Copy the Default Configuration File: Copy the sample configuration file to the `/etc` directory and rename it.

```
sudo cp /opt/odoo18/debian/odoo.conf /etc/odoo18.conf
```

2. Edit the Configuration File: Open the configuration file in a text editor to customize it.

```
sudo nano /etc/odoo18.conf
```

3. Modify the Configuration File: Update the configuration file with the following settings. Be sure to replace the placeholders with your actual values:

```
[options]
; This is the password that allows database operations:
; admin_passwd = admin
db_host = localhost
db_port = 5432
db_user = odoo18
db_password = 123456
addons_path = /opt/odoo18/addons
default_productivity_apps = True
logfile = /var/log/odoo/odoo18.log
```

- db\_host: Set this to `localhost` if the database is on the same server.
- db\_user: The PostgreSQL user you created for Odoo.
- db\_password: The password for the PostgreSQL user.
- addons\_path: The path to the Odoo addons directory.
- logfile: Path to the log file for Odoo.

4. Set File Permissions: Change the ownership and permissions of the configuration file to secure it

```
sudo chown odoo18: /etc/odoo18.conf
```

```
sudo chmod 640 /etc/odoo18.conf
```

5. Create a Log Directory: Create a directory for storing Odoo logs and set the correct permissions.

```
sudo mkdir /var/log/odoo
```

```
sudo chown odoo18:root /var/log/odoo
```

With these steps, you've set up the configuration file with the necessary settings for Odoo 18. You're now ready to proceed with starting the Odoo service and further configurations.

To manage Odoo as a service on your Ubuntu server, you need to create a systemd service file. This will allow you to start, stop, and enable Odoo to run on boot.

Create the Service File: Open a new systemd service file for Odoo 18.

```
sudo nano /etc/systemd/system/odoo18.service
```

Add the Service Configuration: Paste the following content into the service file. This configuration specifies how the Odoo service should run.

```
[Unit]
Description=Odoo18
Documentation=http://www.odoo.com
[Service]
# Ubuntu/Debian convention:
Type=simple
User=odoo18
ExecStart=/opt/odoo18/venv/bin/python3.12 /opt/odoo18/odoo-bin -c /etc/odoo18.conf
[Install]
WantedBy=default.target
```

Type=simple: The service type, simple means that the service will run in the foreground.

User=odoo18: Runs the service as the odoo18 user.

ExecStart: Specifies the command to start Odoo, pointing to the Odoo binary and the configuration file.

Set Permissions for the Service File: Secure the service file by setting the appropriate permissions and ownership.

```
sudo chmod 755 /etc/systemd/system/odoo18.service
```

```
sudo chown root: /etc/systemd/system/odoo18.service
```

Start the Odoo Service: Start the Odoo service using systemctl.

```
sudo systemctl start odoo18.service
```

## Access Odoo in Your Browser: Open your browser and navigate to:

```
http://<your_domain_or_IP_address>:8069
```

```
sudo tail -f /var/log/odoo/odoo18.log
```

Enable the Odoo Service at Boot: Ensure that Odoo starts automatically when the server boots up.

```
sudo systemctl enable odoo18.service
```

Restart the Odoo Service: Apply any changes you've made by restarting the Odoo service.

```
sudo systemctl restart odoo18.service
```