comment installer-odoo-16-erp-software sur-debian-12

Odoo (formerly known as OpenERP) is a self-hosted suite of over 10,000 open-source applications suited for various business needs, including CRM, eCommerce, accounting, inventory, project management, and point of sale. These applications are fully integrated and accessed through a common web interface.

This tutorial will show how to install Odoo 16 on a Debian 12 server.

Prerequisites

1. A Debian 12 based server with a minimum of 2GB RAM to host Odoo Stack.
2. A second Debian 12 based server with a minimum of 2GB RAM for hosting the PostgreSQL database. You can however install the database on the same server as Odoo but for production environments, it is highly recommended that you install it on a separate server. You can also choose any of the managed database options available from any provider of your choice.
3. RAM requirement will depend on the number of concurrent users that will be using the stack. A detailed guide on how to calculate system requirements can be found in Odoo's documentation.
4. Keep your systems updated.

   ```
   $ sudo apt update
   $ sudo apt upgrade
   ```

5. A non-root user with sudo privileges on both servers.
6. Few packages that your systems need.

   ```
   $ sudo apt install wget curl nano ufw software-properties-common dirmngr apt-transport-https gnupg2 ca-certificates lsb-release debian-archive-keyring unzip -y
   ```

   Some of these packages may already be installed on your system.

Step 1 - Configure Firewall rules

For the purpose of this tutorial, we will assume you have `ufw` firewall installed on both servers.

Check the status of the firewall.

```
$ sudo ufw status
```

You should see something like the following.

```
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

On the Odoo server, we will need ports 22, 80, 443, 6010, 5432, and 8069 to be open. 22 is used for SSH, 80 is for HTTP, 443 is for HTTPS, 6010 is used for Odoo communication, 5432 is used by PostgreSQL and 8069 is used for Odoo server application.

Run the following commands to open the required ports on the Odoo server.

```
$ sudo ufw allow 6010,5432,8069,8072/tcp
$ sudo ufw allow http
$ sudo ufw allow https
```

Check the status of the firewall.

```
$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
443                        ALLOW       Anywhere
5432,6010,8069,8072/tcp    ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
443 (v6)                   ALLOW       Anywhere (v6)
5432,6010,8069,8072/tcp (v6) ALLOW     Anywhere (v6)
```

On the PostgreSQL server, we need to open ports 22, 6010, and 5432. Open them using the following commands.

```
$ sudo ufw allow 6010/tcp
$ sudo ufw allow 5432/tcp
```

Step 2 - Assign Hostnames

You can either use the IP addresses of the servers or use their Fully Qualified Domain Names (FQDN), if available. For our tutorial, we will be using FQDNs and for that, we need to set hostnames on both servers.

On the Odoo server, open the `/etc/hosts` file.

```
$ sudo nano /etc/hosts
```

Make sure, it looks like the following.

```
127.0.0.1       localhost
127.0.0.1       odoo.yourdomain.com             odoo
10.1.1.10       postgresql.yourdomain.com       postgresql
```

On the PostgreSQL server, open the file and make sure it looks like the following.

```
127.0.0.1       localhost
127.0.0.1       postgresql.yourdomain.com       postgresql
10.1.2.10       odoo.yourdomain.com             odoo
```

Press **Ctrl + X** to close the editor and press **Y** when prompted to save the file.

Step 3 - Install and Configure PostgreSQL

Debian 12 ships with PostgreSQL 15 by default and we will install that. Run the following command on the PostgreSQL server.

```
$ sudo apt install postgresql-15 postgresql-server-dev-15
```

Next, we need to create a database user `odoo`. You will be asked for a password for the role. Enter a strong password of your choice.

```
$ sudo -u postgres createuser odoo -U postgres -dP
```

The option `-u` executes the command as `postgres` user.

The option `-U` indicates the user name to connect as.

The option `-d` grants the user permission to create databases.

The option `-p` prompts for the new user's password.

Configure Host-Based Authentication

We need to give permission to the PostgreSQL service to be able to connect to the Odoo server.

First, stop the PostgreSQL service.

```
$ sudo systemctl stop postgresql
```

Open the file `/etc/postgresql/15/main/pg_hba.conf` for editing.

```
$ sudo nano /etc/postgresql/15/main/pg_hba.conf
```

Paste the following line at the end.

```
host            all             odoo            odoo.yourdomain.com             md5
```

This line grants permission to the `odoo` user to connect to all the databases within this server. You can specify the name of the databases too instead of using the `all` keyword.

Press **Ctrl + X** to close the editor and press **Y** when prompted to save the file.

Configure PostgreSQL Listening address

Next, we need to allow the database server to listen to remote connections. Open the file `/etc/postgresql/15/main/postgresql.conf` for editing.

```
$ sudo nano /etc/postgresql/15/main/postgresql.conf
```

Change the line `listen_addresses` from

```
#listen_addresses = 'localhost' # what IP address(es) to listen on;
```

to.

```
#From CONNECTIONS AND AUTHENTICATION Section
listen_addresses = '*'
```

The `*` means it will listen to all the IP addresses. You can change it to the IP address of your odoo instance.

Press **Ctrl + X** to close the editor and press **Y** when prompted to save the file.

Enable and Start the PostgreSQL service

Since our configuration is finished, it is time to start and enable the PostgreSQL service.

```
$ sudo systemctl enable postgresql --now
```

Step 4 - Install Odoo

Install dependencies and Prepare for installation

Create a new system user for managing the Odoo processes on the Odoo server.

```
$ sudo adduser --system --home=/opt/odoo --group odoo
```

Install system dependencies required for Odoo 16 setup.

```
 $ sudo apt install python3-pip python3-suds python3-all-dev python3-venv python3-dev python3-setuptools python3-tk libxml2-dev libxslt1-dev libevent-dev libsasl2-dev libldap2-dev pkg-config libtiff5-dev libjpeg62-turb
```

Install Nodejs. Debian 12 ships with Node 18.x which is the current LTS version of Nodejs. However, we will use the official Nodesource repository for it.

Download and import the Nodesource GPG key.

```
$ sudo apt-get install -y ca-certificates curl gnupg
$ sudo mkdir -p /etc/apt/keyrings
$ curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg
```

Create the system repository.

```
$ NODE_MAJOR=18
$ echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list
```

Update the system repositories list.

```
$ sudo apt update
```

Install Nodejs.

```
$ sudo apt install nodejs
```

Confirm the Node version.

```
$ node --version
v18.17.1
```

Install Less CSS package using Node. If you are using a language with right-to-left interface, install the `rtlcss` package as well.

```
$ sudo npm install -g less rtlcss
```

Install `wkhtmltopdf`. It is available as a Debian system package.

```
$ sudo apt install wkhtmltopdf
```

Check the version of `wkhtmltopdf` installed.

```
$ wkhtmltopdf --version
wkhtmltopdf 0.12.6
```

Download Odoo Files

Clone Odoo's Github repository onto your system.

```
$ sudo git clone https://github.com/odoo/odoo.git --depth 1 --branch 16.0 --single-branch /opt/odoo
```

For our purpose, we are copying Odoo to the `/opt/odoo` directory from where it will be installed.

Set up Virtualenv Python Environment

This step is optional but is recommended since a virtual Python environment for Odoo will help avoid conflicts with Python modules of the Operating system, especially when performing OS upgrades.

For this, we will use `virtualenv`.

1. Create a new `virtualenv` environment for Odoo.

   ```
   $ python3 -m venv /home/<username>/odoo-env
   ```

2. Activate the virtual environment. We are creating an environment under the system user's home directory. You are free to choose any location you like.

   ```
   $ source /home/<username>/odoo-env/bin/activate
   ```

3. Update PIP just in case.

   ```
   (odoo-env) $ pip3 install --upgrade pip
   ```

4. Install Python's wheel in the virtual environment.

   ```
   $ pip3 install wheel
   ```

Install Python Dependencies

Install the Python dependencies required by Odoo 16.

```
$ pip3 install -r /opt/odoo/requirements.txt
```

The requirements will take some time to install so be patient.

Check whether the requirements are installed correctly by checking the list of installed Python modules.

```
$ pip3 list
Package            Version
------------------ ---------
appdirs            1.4.4
attrs              23.1.0
Babel              2.9.1
beautifulsoup4     4.12.2
cached-property    1.5.2
certifi            2023.7.22
cffi               1.15.1
chardet            4.0.0
cryptography       3.4.8
decorator          4.4.2
defusedxml         0.7.1
docopt             0.6.2
docutils           0.16
ebaysdk            2.1.5
freezegun          0.3.15
gevent             22.10.2
greenlet           2.0.2
idna               2.10
isodate            0.6.1
Jinja2             3.1.2
libsass            0.20.1
lxml               4.9.2
MarkupSafe         2.1.2
num2words          0.5.9
ofxparse           0.21
passlib            1.7.4
Pillow             9.4.0
pip                23.2.1
polib              1.1.0
psutil             5.9.4
psycopg2           2.9.5
pyasn1             0.5.0
pyasn1-modules     0.3.0
pycparser          2.21
pydot              1.4.2
pyOpenSSL          20.0.1
pyparsing          3.1.1
PyPDF2             2.12.1
pyserial           3.5
python-dateutil    2.8.1
python-ldap        3.4.0
python-stdnum      1.16
pytz               2023.3
pyusb              1.2.1
qrcode             6.1
reportlab          3.6.12
requests           2.25.1
requests-file      1.5.1
requests-toolbelt  1.0.0
setuptools         66.1.1
six                1.16.0
soupsieve          2.5
urllib3            1.26.5
vobject            0.9.6.1
Werkzeug           2.0.2
wheel              0.41.2
xlrd               1.2.0
XlsxWriter         1.1.2
xlwt               1.3.0
zeep               4.0.0
zope.event         5.0
zope.interface     6.0
```

Exit the Python virtual environment.

```
$ deactivate
```

Install Python Packages

Odoo 16 requires a few more Python packages which are listed in the `/opt/odoo/debian/control` file. Switch to the `/opt/odoo` directory.

```
$ cd /opt/odoo
```

Run the following command to install the required Python packages.

```
$ sed -n -e '/^Depends:/,/^Pre/ s/ python3-\(.*\),/python3-\1/p' debian/control | sudo xargs apt-get install -y
```

Step 5 - Configure Odoo

Copy the default Odoo configuration file to create a new one.

```
$ sudo cp /opt/odoo/debian/odoo.conf /etc/odoo-server.conf
```

Open the file for editing.

```
$ sudo nano /etc/odoo-server.conf
```

Edit the file so that it looks like the following.

```
[options]
; This is the password that allows database operations:
admin_passwd = admin
db_host = postgresql.yourdomain.com
db_port = False
db_user = odoo
db_password = odoo_password
addons_path = /opt/odoo/addons
xmlrpc_port = 8069
default_productivity_apps = True
```

Press **Ctrl + X** to close the editor and press **Y** when prompted to save the file.

The option `admin_passwd` is the password that allows administrative operations within the Odoo GUI. Be sure to choose a secure password.

The option `db_host` is the FQDN or the IP address of the PostgreSQL server.

The option `db_port` is set to false since the default PostgreSQL port 5432 is being used. If you want to use a different port, you will need to update this value.

The option `db_user` is the name of the PostgreSQL user.

The option `db_password` is the PostgreSQL 'odoo' user password we created previously on the PostgreSQL server.

The option `addons_path` is the default Addons path. You can also add a custom path for Addons separating them with commas.

The option `xmlrpc_port` is the port that Odoo listens on.

The option `default_productivity_apps` makes sure the default productivity apps (namely Employees, Email Marketing, Project, and Surveys) remain enabled. These four apps are the default on the Odoo Community Edition. On the Enterprise edition, there are additional productivity apps that can be enabled by default which are Appointments, Knowledge, Planning, and Sign.

Create Odoo service

To make sure Odoo keeps running even after a system restarts, we need to create a service for it.

Create a file `/lib/systemd/system/odoo-server.service` and open it for editing.

```
$ sudo nano /lib/systemd/system/odoo-server.service
```

Paste the following code in it.

```
[Unit]
Description=Odoo Open Source ERP and CRM
After=network.target

[Service]
Type=simple
SyslogIdentifier=odoo-server
User=odoo
Group=odoo
ExecStart=python3 /opt/odoo/odoo-bin --config=/etc/odoo-server.conf --addons-path=/opt/odoo/addons/ --logfile /var/log/odoo/odoo-server.log
WorkingDirectory=/opt/odoo/
StandardOutput=journal+console
KillMode=mixed

[Install]
WantedBy=multi-user.target
```

Replace `/home/<username>` with the location you chose for installing the Python Virtual Environment.

Press **Ctrl + X** to close the editor and press **Y** when prompted to save the file.

Create a Log directory for Odoo

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

Set File permissions

Set permissions on the `odoo-server.service` file so that only Odoo users can read or execute it.

```
$ sudo chmod 755 /lib/systemd/system/odoo-server.service
$ sudo chown odoo: /lib/systemd/system/odoo-server.service
```

Set the ownership on the Python environment, the Odoo installation, and the log directory.

```
$ sudo chown -R odoo: /opt/odoo/
$ sudo chown -R odoo: /home/<username>/odoo-env
$ sudo chown -R odoo: /var/log/odoo
```

Restrict the Odoo configuration file.

```
$ sudo chown odoo: /etc/odoo-server.conf
$ sudo chmod 640 /etc/odoo-server.conf
```

Start the Odoo server

Start and enable the Odoo server.

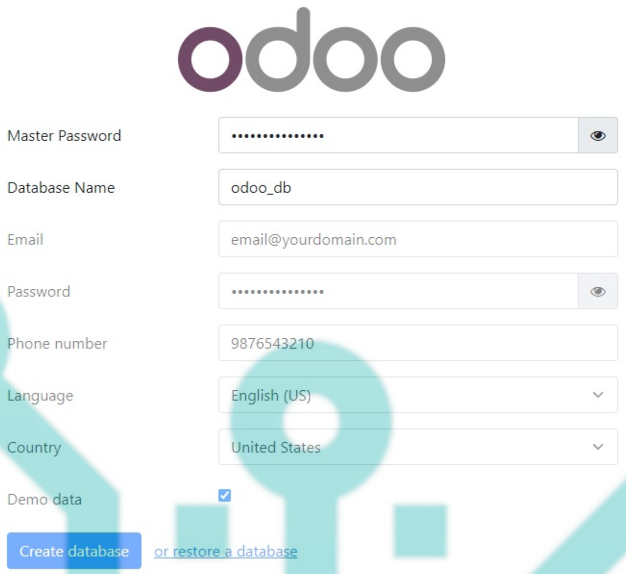
```
$ sudo systemctl start odoo-server
$ sudo systemctl enable odoo-server
```

Check the status of the server.

```
$ sudo systemctl status odoo-server
? odoo-server.service - Odoo Open Source ERP and CRM
     Loaded: loaded (/lib/systemd/system/odoo-server.service; enabled; preset: enabled)
     Active: active (running) since Mon 2023-09-04 14:19:38 UTC; 27s ago
   Main PID: 8954 (python3)
      Tasks: 4 (limit: 2315)
     Memory: 131.1M
        CPU: 1.625s
     CGroup: /system.slice/odoo-server.service
             ??8954 python3 /opt/odoo/odoo-bin --config=/etc/odoo-server.conf --addons-path=/opt/odoo/addons/ --logfile /var/log/odoo/odoo-server.log

Sep 04 14:19:38 odoo systemd[1]: Started odoo-server.service - Odoo Open Source ERP and CRM.
```

In your browser, open the URL `http://<yourIPaddress>:8069` or `http://odoo.yourdomain.com:8069`. If everything is working properly, you should see Odoo's database creation screen.



Master Password ••••••••••••••

Database Name odoo_db

Email email@yourdomain.com

Password ••••••••••••••

Phone number 9876543210

Language English (US)

Country United States

Demo data ☑

Fill in all the fields. Check the `Demo Data` field to populate the database with sample data and then click the **Create database** button.

Next, you will be shown a list of apps that you can choose and select.

The first time you create a database, the addons page will take time to load so don't refresh the page.

Step 6 - Install and Configure Nginx

Debian 12 ships with an older version of Nginx. To install the latest version, you need to download the official Nginx repository.

Import Nginx's signing key.

```
$ curl https://nginx.org/keys/nginx_signing.key | gpg --dearmor \
    | sudo tee /usr/share/keyrings/nginx-archive-keyring.gpg >/dev/null
```

Add the repository for Nginx's stable version.

```
$ echo "deb [signed-by=/usr/share/keyrings/nginx-archive-keyring.gpg] \
http://nginx.org/packages/debian `lsb_release -cs` nginx" \
    | sudo tee /etc/apt/sources.list.d/nginx.list
```

Update the system repositories.

```
$ sudo apt update
```

Install Nginx.

```
$ sudo apt install nginx
```

Verify the installation. On Debian systems, the following command will only work with `sudo`.

```
$ sudo nginx -v
nginx version: nginx/1.24.0
```

Start Nginx.

```
$ sudo systemctl start nginx
```

Check the service status.

```
$ sudo systemctl status nginx
? nginx.service - nginx - high performance web server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Tue 2023-09-05 06:29:17 UTC; 1s ago
       Docs: https://nginx.org/en/docs/
    Process: 13958 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf (code=exited, status=0/SUCCESS)
   Main PID: 13959 (nginx)
      Tasks: 2 (limit: 2315)
     Memory: 1.7M
        CPU: 10ms
     CGroup: /system.slice/nginx.service
             ??13959 "nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf"
             ??13960 "nginx: worker process"
```

Step 7 - Install SSL

We need to install Certbot to generate free SSL certificates offered by Let's Encrypt.

You can either install Certbot using Debian's repository or grab the latest version using the Snapd tool. We will be using the Snapd version.

Debian 12 comes doesn't come with Snapd installed. Install Snapd package.

```
$ sudo apt install snapd
```

Run the following commands to ensure that your version of Snapd is up to date.

```
$ sudo snap install core
$ sudo snap refresh core
```

Install Certbot.

```
$ sudo snap install --classic certbot
```

Use the following command to ensure that the Certbot command can be run by creating a symbolic link to the `/usr/bin` directory.

```
$ sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

Verify if Certbot is functioning properly.

```
$ certbot --version
certbot 2.6.0
```

Generate the certificate. We also need to create a DHParams certificate.

```
$ sudo certbot certonly --nginx --agree-tos --no-eff-email --staple-ocsp --preferred-challenges http -m email@yourdomain.com -d odoo.yourdomain.com
```

Generate a Diffie-Hellman Key-exchange certificate.

```
$ sudo openssl dhparam -dsaparam -out /etc/ssl/certs/dhparam.pem 4096
```

Check the Certbot renewal scheduler service.

```
$ sudo systemctl list-timers
```

You will find `snap.certbot.renew.service` as one of the services scheduled to run.

```
NEXT                        LEFT          LAST                        PASSED        UNIT                      ACTIVATES
.....
Sun 2023-02-26 06:32:00 UTC 9h left       Sat 2023-02-25 18:04:05 UTC 2h 59min ago  snap.certbot.renew.timer  snap.certbot.renew.service
Sun 2023-02-26 06:43:20 UTC 9h left       Sat 2023-02-25 10:49:23 UTC 10h ago       apt-daily-upgrade.timer   apt-daily-upgrade.service
Sun 2023-02-26 09:00:06 UTC 11h left      Sat 2023-02-25 20:58:06 UTC 5min ago      apt-daily.timer           apt-daily.service
```

Do a dry run of the process to check whether the SSL renewal is working fine.

```
$ sudo certbot renew --dry-run
```

If you see no errors, you are all set. Your certificate will renew automatically.

Step 8 - Configure Nginx

To run it via Nginx, we need to run Odoo on localhost. To change that, stop the Odoo service.

```
$ sudo systemctl stop odoo-server
```

Open the Odoo server configuration file.

```
$ sudo nano /etc/odoo-server.conf
```

Add the following lines to it.

```
xmlrpc_interface = 127.0.0.1
proxy_mode = True
```

Create an Nginx configuration file for Odoo.

```
$ sudo nano /etc/nginx/conf.d/odoo.conf
```

Paste the code below.

```
#odoo server
upstream odoo {
 server 127.0.0.1:8069;
}
upstream odoochat {
 server 127.0.0.1:8072;
}

# http -> https
server {
   listen 80;
   listen [::]:80;
   server_name odoo.yourdomain.com;
   return 301 https://$host$request_uri;
}

server {
 listen 443 ssl http2;
 listen [::]:443 ssl http2;
 server_name odoo.yourdomain.com;

 proxy_read_timeout 720s;
 proxy_connect_timeout 720s;
 proxy_send_timeout 720s;

 # Add Headers for odoo proxy mode
 proxy_set_header X-Forwarded-Host $host;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto $scheme;
 proxy_set_header X-Real-IP $remote_addr;

 # SSL parameters
 ssl_certificate /etc/letsencrypt/live/odoo.yourdomain.com/fullchain.pem;
 ssl_certificate_key /etc/letsencrypt/live/odoo.yourdomain.com/privkey.pem;
 ssl_trusted_certificate /etc/letsencrypt/live/odoo.yourdomain.com/chain.pem;

 ssl_session_timeout 1d;
 ssl_session_cache shared:MozSSL:10m;  # about 40000 sessions
 ssl_session_tickets off;

 ssl_protocols TLSv1.2 TLSv1.3;
 ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-R
 ssl_ecdh_curve X25519:prime256v1:secp384r1:secp521r1;
 ssl_prefer_server_ciphers off;
 ssl_stapling on;
 ssl_stapling_verify on;
 ssl_dhparam /etc/ssl/certs/dhparam.pem;

 # log
 access_log /var/log/nginx/odoo.access.log;
 error_log /var/log/nginx/odoo.error.log;

 # Redirect longpoll requests to odoo longpolling port
 location /longpolling {
 proxy_pass http://odoochat;
 }

 # Redirect requests to odoo backend server
 location / {
   proxy_redirect off;
   proxy_pass http://odoo;
 }

 # common gzip
 gzip_types text/css text/scss text/plain text/xml application/xml application/json application/javascript;
 gzip on;
}
```

Press **Ctrl + X** to close the editor and press **Y** when prompted to save the file.

Open the file `/etc/nginx/nginx.conf` for editing.

```
$ sudo nano /etc/nginx/nginx.conf
```

Add the following line before the line `include /etc/nginx/conf.d/*.conf;`.

```
server_names_hash_bucket_size  64;
```

Save the file by pressing **Ctrl + X** and entering **Y** when prompted.

Test the Nginx configuration.

```
$ sudo nginx -t
```

If you see no errors, it means you are good to go. Restart the Nginx server.

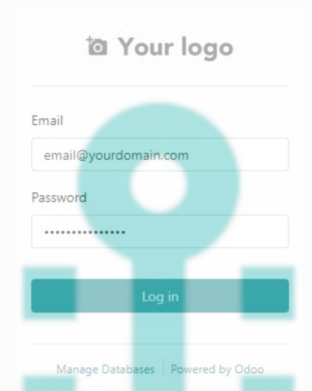
```
$ sudo systemctl restart nginx
```

Step 9 - Start Odoo

Now that everything is set up, we can start the Odoo server again.

```
$ sudo systemctl start odoo-server
```

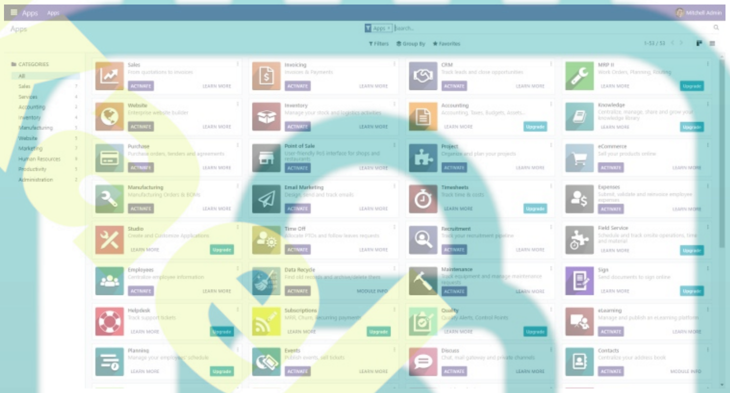
Launch Odoo in your browser via `https://odoo.yourdomain.com`. You will get a screen described earlier. Enter the required details to create the database and you will be redirected to the Odoo login page.



Email: email@yourdomain.com

Password: ••••••••••••••••

Enter the credentials used in the previous step to log in to the Odoo ERP and you will be taken to the dashboard page as shown below. It will be pre-filled with demo data. If you haven't checked the demo data option on the database creation page, you will get a different and much cleaner dashboard.



You can start using Odoo to manage your business from here on.

Conclusion

This concludes our tutorial on installing Odoo on the Debian 12 server. If you have any questions, post them in the comments below.