Odoo SaaS Kit

OverView

Once you purchase the module, you will get the odoo saas kit.
1.) odoo_saas_kit : Main addon folder to be installed at the server end.

After purchasing the module please raise a request with our support team to get the other following needed modules from us. If you have purchased the installation we will install the modules from our side.
2.) wk_saas_tool : Module needed to allow user to login to the clients & templates containers directly from SAAS kit panel. And will be installed at client's end.

Our Odoo SaaS Kit module uses Docker to create unique Odoo SaaS Instance containers for each of your Clients on the designated server.

The instance is the container of the docker. You can also decide whether you wish to deploy it on same server or a remote server. Docker Community Edition is used to make this work.

Before Installing the Module in your Odoo

First install the following python libraries to your system.

- docker (For Docker Python APIs)
- erppeek (For Odoo managed Tasks)
- paramiko (For remote SSH connections) In case you plan to have remote containers.

Use the following command in your command prompt to install the libraries >>
pip3 install docker erppeek paramiko

Installation & Setup of Odoo SaaS Kit

1) Place the Odoo_SaaS_Kit folder in appropriate addons path folder.
2) Update the Odoo app list in Odoo.
3) Install the module from Odoo Backend.
4.) Once successfully installed. Next step is to create few needed folders:

a.) Odoo-SAAS-Data:
This folder is created in the Odoo_SaaS_Kit folder. The folder will keep all the necessary files & folder for SAAS clients e.g Client's odoo.conf , their data directory and Nginx Vhosts ,etc.

b.) docker_vhosts
The folder should be inside Odoo-SAAS-Data & will contain the Nginx Virtual hosts files for clients in order to have sub-domains per client.

c.)common_addons
The folder will be shared amongst all the Clients & templates as an addons path. The modules placed in this directory will be visible to all Clients & Templates. (This folder can be created anywhere in the system where Odoo is installed.)

Note: The ownership of all created folders in the above steps should be given to the same User from which the Odoo is running.

5.) Place the following files in the appropriate folders under the paths as shown below:

a.) Odoo-SAAS-Data/odoo.conf
The file is default Odoo configuration file for SAAS clients. This file will be used as a reference at the time of Client container creation.

b.) Odoo-SAAS-Data/odoo-template.conf
The file is default Odoo configuration file for SAAS template. This file will be used as a reference at the time of Template container creation.

c.) Odoo-SAAS-Data/docker_vhosts/vhosttemplate.txt
The default template for Nginx vhosts

Creating A Base Odoo Docker Image

You need to create a base Odoo image to be used by all Odoo SaaS clients.

First, you need to install the docker setup in your system.
1.) Visit https://docs.docker.com/install/linux/docker-ce/ubuntu/ to know how to install the docker-ce.
2. Once installed, generate a docker image that will be used to create all the containers.

Note:- All the files are available in the provided zip file

Please keep the following points in mind while creating the Base Docker image.

A.) User ID & Group ID of Odoo Service User should be same on Host machine & Docker image:

The odoo_saas_kit module maintains data directory(which includes filestore & sessions) , odoo configuration file and common addons on Host i.e. outside the containers. Appropriate paths are created & mounted for each client at runtime. Since the files are being shared , these shared folders need to have proper ownership on Host & inside docker containers. So, one needs to ensure the owner (on Host and in containers) have the same user id & group id.

B.) You need to update the User ID & Group ID in Dockerfile to match the same of odoo Service user on App server.
Refer to /etc/passwd file to check the userid & groupid of the Odoo service user.

C.) Odoo service user should own Dockerfile and other files kept adjacent to Dockerfile.
There are some executable files which should be owned by Odoo user in order to run Odoo application inside containers.

Note:- In case you plan to use remote containers, you would need to configure docker daemon to listen publicly.

Command: -
cd {your_docker_file_path}
docker build -t {your_docker_image_name:your_docker_image_tag} .

Please add the period or dot I.e “.” at the end of above command.

Additional Permissions To Be Given To The Odoo Service User

You need to provide additional permissions to the Odoo Service User

A.) Allow Odoo user to do docker commands
Because the docker will be controlled by Odoo application now.
Command- usermod -a -G docker {your_odoo_user}

B.) Allow odoo user to control Nginx as the odoo_saas_kit module will configure Nginx Vhosts at runtime.
Command- Append “{your_odoo_user} ALL=(ALL) NOPASSWD:/usr/sbin/nginx” to /etc/sudoers

C.) Include the Nginx conf location for SAAS
The module maintains all Nginx Vhosts here. So, including the folder in main Nginx configuration will allow dynamically generated Vhosts to have effect.
Add “include {your_docker_vhosts_path}/*.conf;” in your main Nginx Server Block.

D.) Enable Connectivity with Postgresql Server because the clients & templates will need a Postgresql server. Ensure the connectivity to that Postgresql Server along with login & db creation permissions.
Update postgresql.conf & pg_hba.conf accordingly

E.) Create a new user for SAAS with appropriate user permissions & a password to have a separate Odoo user for all SAAS clients.

FInally ensure that the following paths are properly configured in saas.conf (in odoo_saas_kit/models/lib/)

1.) nginx_vhosts(e.g. /opt/odoo/Odoo-SAAS-Data/docker_vhosts/)
Path where Nginx vhosts will be created.

2.) odoo_saas_data(e.g. /opt/odoo/Odoo-SAAS-Data/)
Path where all folders/files related to SAAS Clients will be placed/kept.

3.) common_addons(e.g. /opt/odoo/common_addons)
Path where all common addons will be kept/placed. Module odoo_saas_kit will mount this location to every SAAS clients for the purpose of common addons

4.) odoo_image(e.g. odoobywebkul:12.0)
Docker Image that must be used for SAAS client creation.

5.) template_odoo_port(e.g. 8888)
Port that will be occupied for template container & it should not be occupied.
.

You can add another server i.e remote server to SAAS kit. And choose to create SAAS Clients on that remote Server

Setting up Remote Server for SaaS Kit in order to add it SAAS kit

1.) Create odoo System User.

Note: Odoo system user's uid and gid should match the uid and gid of odoo user on main server.

Command:

adduser --system --home /opt/{your_odoo_user} --shell /bin/bash --uid {odoo_user_uid} --gid {odoo_user_gid} {your_odoo_user}

2.) Add password to the odoo user created and enable Password based SSH authentication:

Command:

passwd {odoo_app_user}

3.) Now, you need to install docker same as Main Server.

Visit https://docs.docker.com/install/linux/docker-ce/ubuntu/ to know how to install the docker-ce.

4.) After docker installation, **export image which was built on Main Server to Remote Server**

Command(On Main Server):-

docker save {odoo_saas_image} > {odoo_saas_image}.tar

Now, Copy tar file from main server to remote server. And load it **on remote server.**

Command:

docker load < {odoo_saas_image}.tar

Alternatively, image can also be pushed to docker hub and can be pulled on remote.

Or, if none of the above options work image can be rebuilt using the same Dockerfile along with other files present in its folder e.g entrypoint.sh, run_odoo.sh etc.

5.) Allow odoo user to use docker by adding odoo user to docker group.

Command:

usermod -a -G docker {your_odoo_user}

6) You also have to make Remote Docker Daemon Listen on Port 2375 instead of Socket by updating the Service file.

Update ExecStart Option to below in file **/lib/systemd/system/docker.service**:-

ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:2375

Reload Systemctl Daemon:

systemctl daemon-reload

Restart Docker Service:

service docker restart

NOTE:- This would expose Docker Daemon and ensure that you configure Firewall Rules to block public traffic on this Port I.e 2375

7.) Now, you need to install few pip3 dependencies.

3. docker (For Docker Python APIs)
4. erppeek (For Odoo managed Tasks)
5. paramiko (For remote SSH connections) In case you plan to have remote containers.
6. psycopg2 (For connecting postgresql using python).

Use the following command in your command prompt to install the libraries >>

pip3 install docker erppeek paramiko psycopg2

Note: You might need to install libpq-dev before installing psycopg2

Command:

sudo apt install libpq-dev

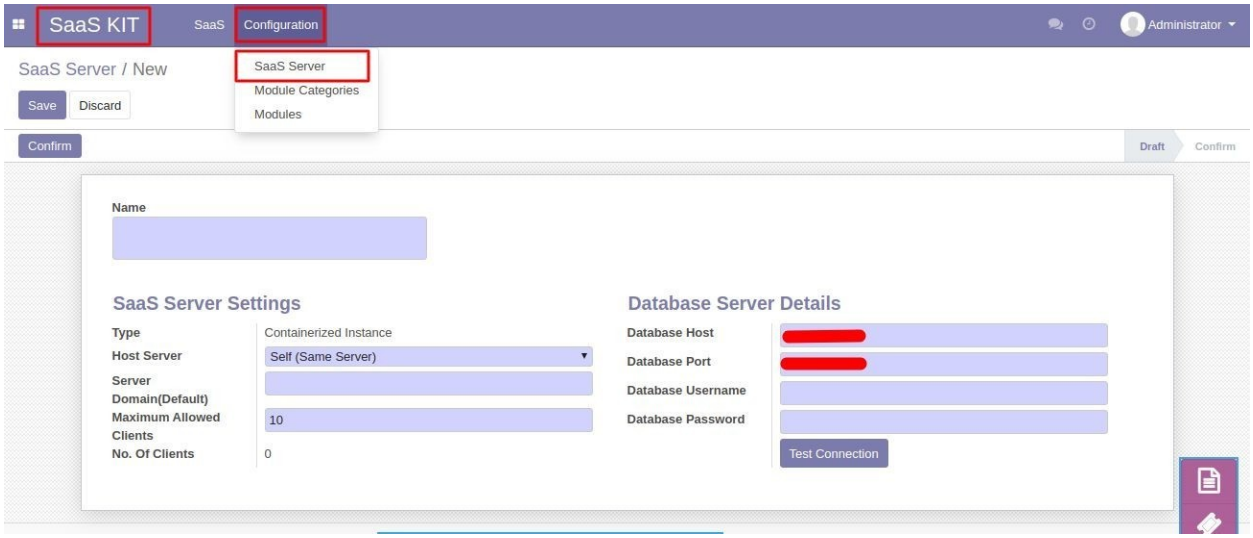
8.) You also need to Create **Odoo-SAAS-Data(just add odoo.conf, odoo-template.conf within it) and common_addons(including all files/folders)** same as main server with **same permissions and ownership.**

Once all steps are performed, you can add this remote server to saas kit.

NOTE:- Both Main & All your Remote Servers are suggested to be in a local n/w i.e locally reachable from each other using Private IPs.

How To Configure SaaS Server

1.) Configure SaaS Kit module server by adding details of your server to it.



2.) Go to SaaS Kit >> Configuration >> SaaS Server

3.) Enter the **Name** of Server. Select the **Host Server**

- Self (for Same Server)
- Remote Server)

4.) Then, enter all the following details in SaaS Server Settings.

a.) Host Server

Select Self(Same Server) as **Host Server** to create and run all the client's instances on the same server which you are using to run your Odoo.

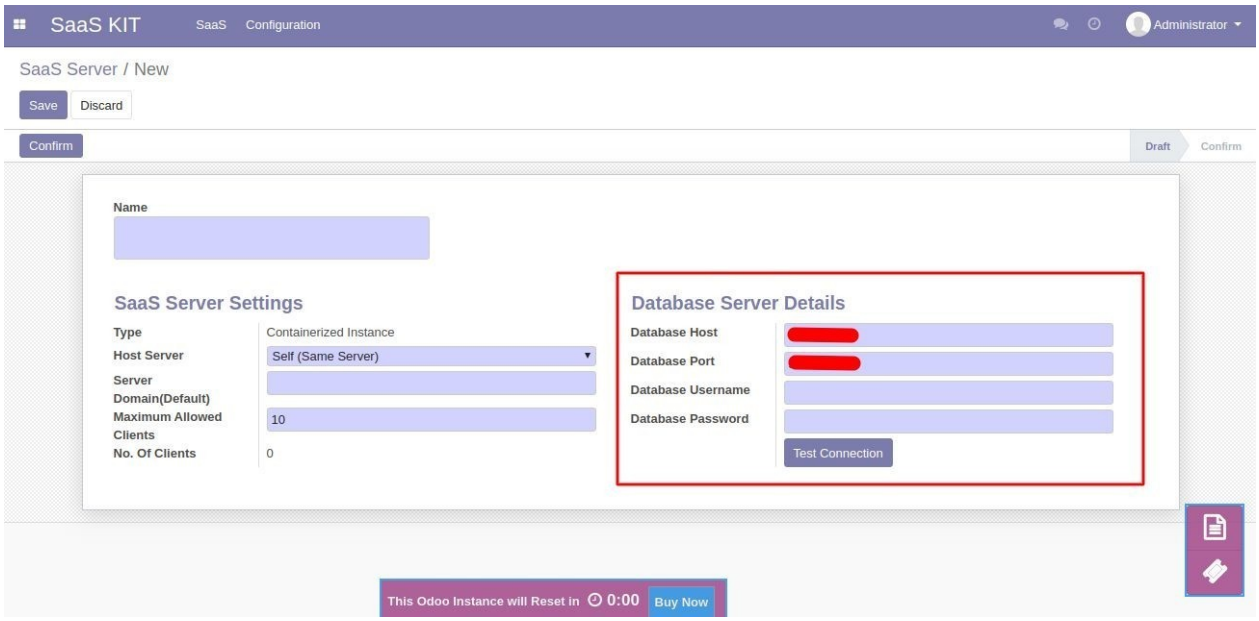
b.) Server Domain(Default)

Enter the Domain name of your server. Ex-(webkul.com, mobikul.com).

c.) Maximum Allowed Client

Set Maximum number of client's instances to be created as per the size and load of your server.

5.) Now, along with the server details you need to enter the database server details to complete the server configuration.



a.) Database Host

Enter the host name on which you database server is running.

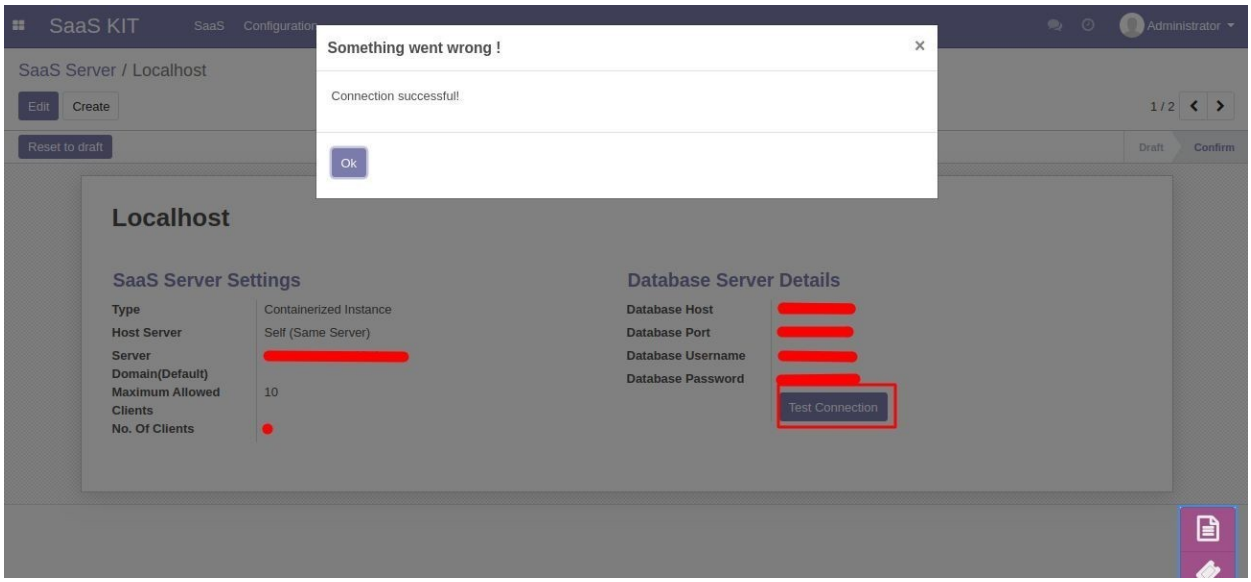
b.) Database Port

Enter the Port number on which database server is listening..

c.) Database Username and Database Password

Enter DB username and password for that username.

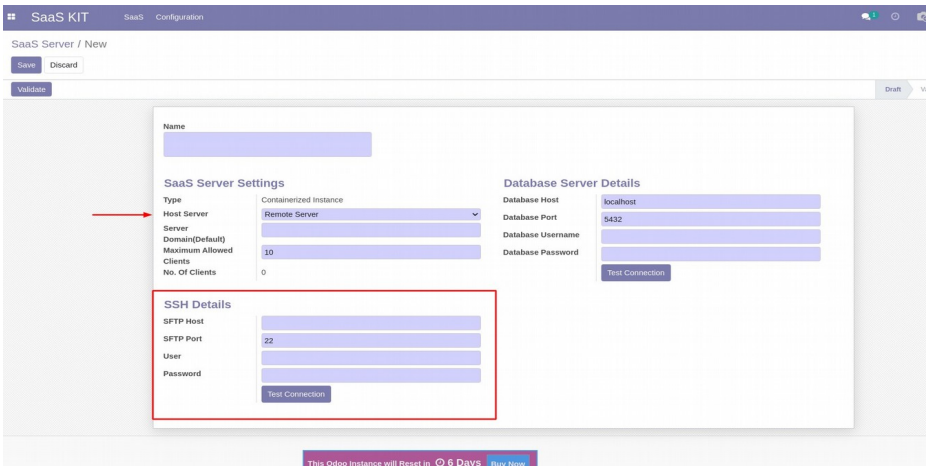
6.) Once you enter the details, you can test the db connection by clicking on **‘Test Connection’**. A pop-up having message **Connection Successful !** will appear on your screen if all the details are correct.



7.)Save the details and click on ‘**Confirm’** button to complete the configuration**.**

How To Configure SaaS Remote Server

1.) Create New configuration for saas server by following the same steps as mentioned previously.



2.) Enter Name of the Server and select **Remote Host** in Host server field.

3.) Then add details in following field:

a.) **Server Domain**

Enter the Domain name of your server. Ex-(webkul.com, mobikul.com).

b.) **Maximum allowed Clients**

Set Maximum number of client's instances to be created as per the size and load of your server.

4.) Now add the SSH details for the remote server.

a.) **SFTP Host**

Enter the local IP address of the remove server.

b.) **SFTP Port**

Enter the port for ssh connection (default 22).

c.) **User and Password**

Enter the username and password for ssh connection.

Also, add database server details as mentioned in previous steps and test DB connection.

Note:- **Never enter localhost here, enter the local/Private IP of the Postgresql Server.**
You have to update Postgresql config to allow connections from 172.17.0.0/16 CIDR.

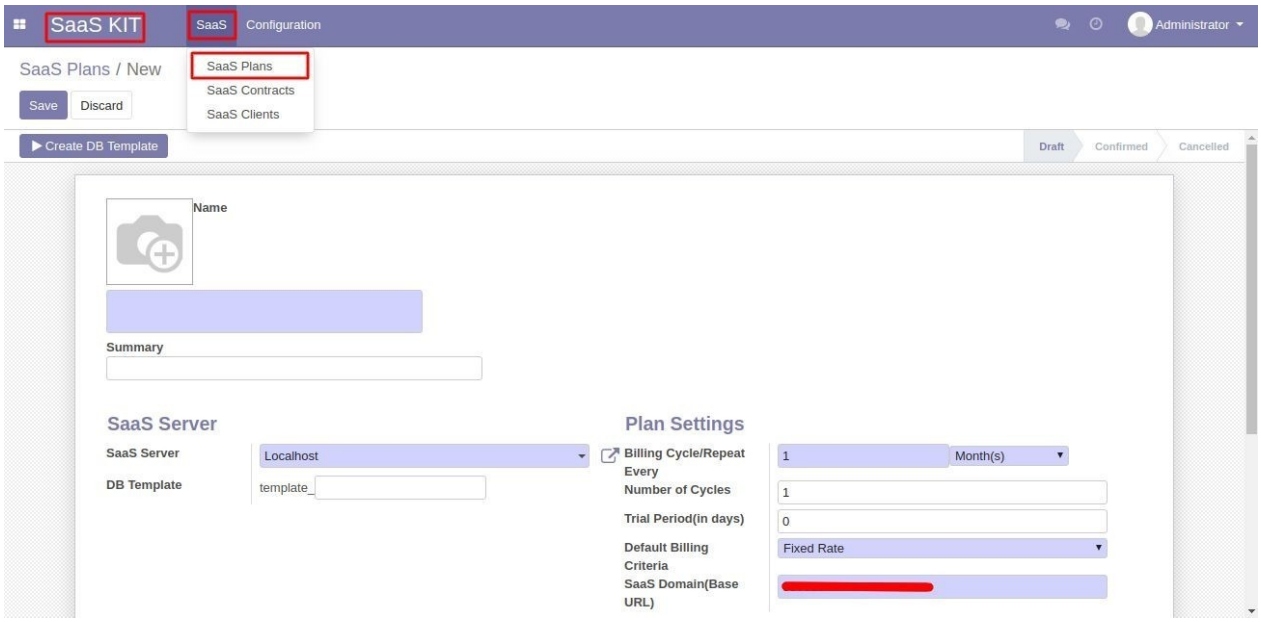
5.) Save Details and Click on confirm button to complete configuration.

If there is any error, a message would pop-up.

Creating SaaS Subscription Plans

1.) The plans would allow the customer to create his/her Odoo instance and use the modules mentioned in the plans for their business.

2.) Go to SaaS Kit >> SaaS >> SaaS Plan



2.) Enter the **Name** of Plan. Add the description for plan in **Summary** if needed.

3.) add the following details:

a.) SaaS Server

Select a server among the list of servers which you have configured and confirmed before, so that odoo's instances created by this plan can run over that particular server.

b.) DB Template

Enter some unique name for db_template of this plan, if you leave this field blank then it will auto assign a name by the name of plan and a unique number.

c.) Billing Cycle/ Repeat Every

Select time period from (Days, Weeks, Months, Year etc) and enter the value for that time period. This field represents the time period of a billing cycle of this plan.

d.) Number Of Cycles

Enter number of billing cycles you want to sell of odoo intances associated to this plan.

e.) Trial Period(in days)

Enter no. of days for trial period. If you enter days then the calculation of billing cycle will be started after the mentioned days.

f.) Default Billing Criteria

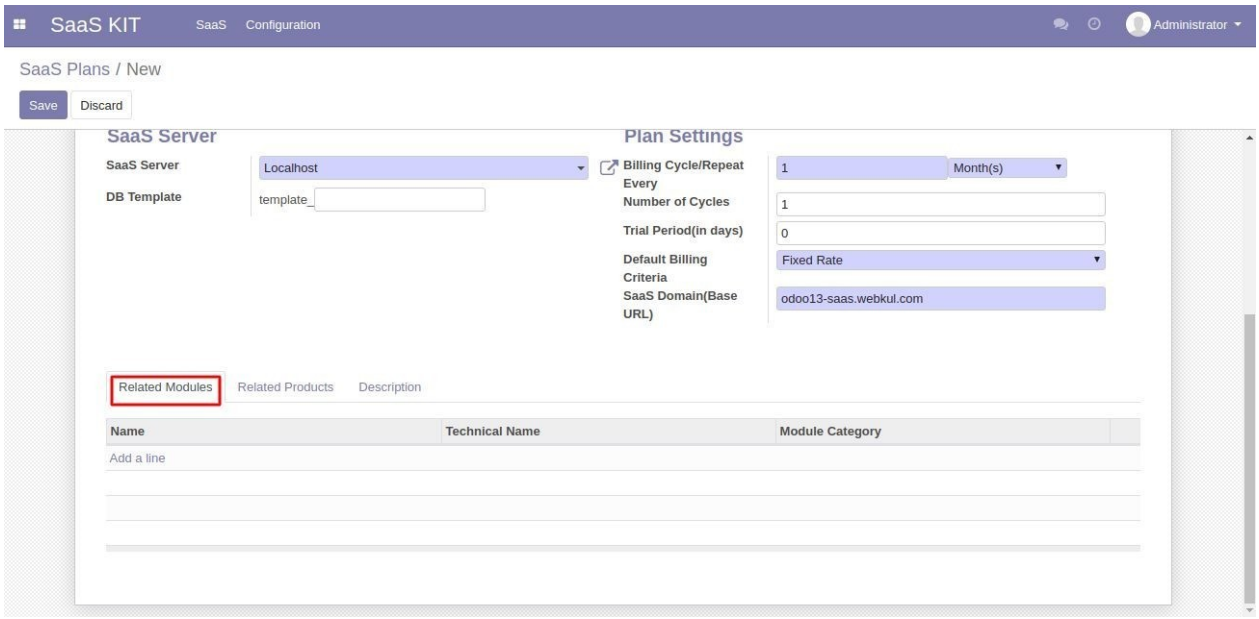
Select Fixed Rate or Based on the No. of users.
For Fixed Rate: Invoice of a fixed amount will be generated after every billing cycle.

For Based on the No. of users: Invoice is generated according to the number of users created on the customer’s instance.

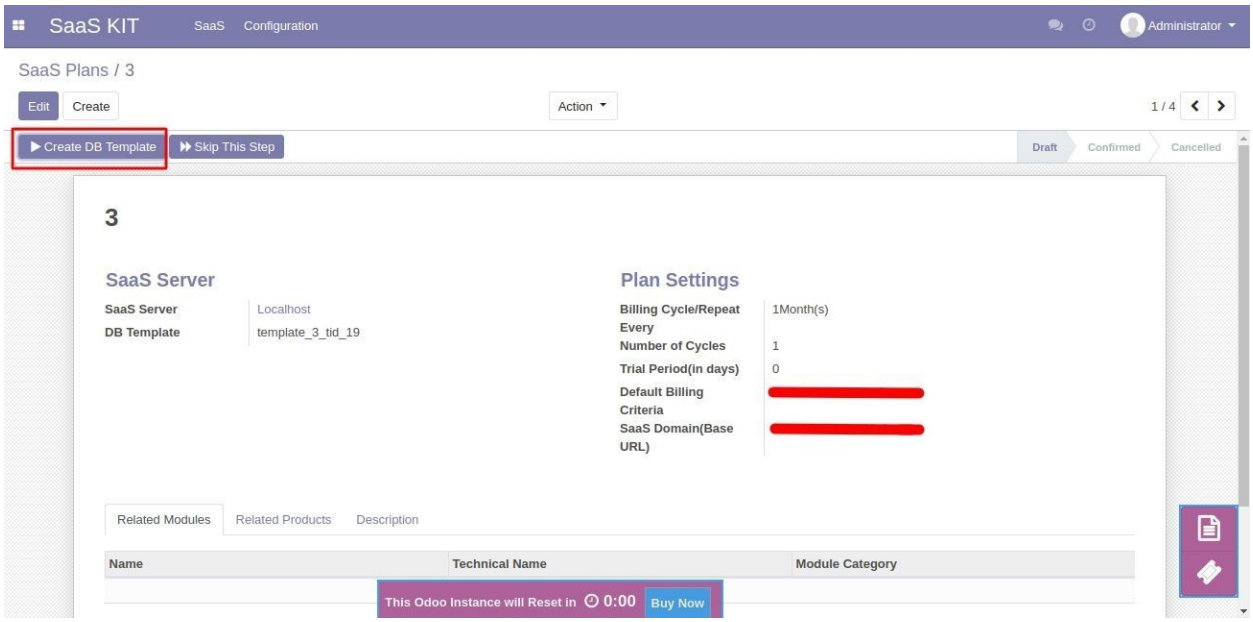
g.) SaaS Domain(Base URL)

Enter domain in this field although it auto filled with the domain name you have enterd in server configuration.

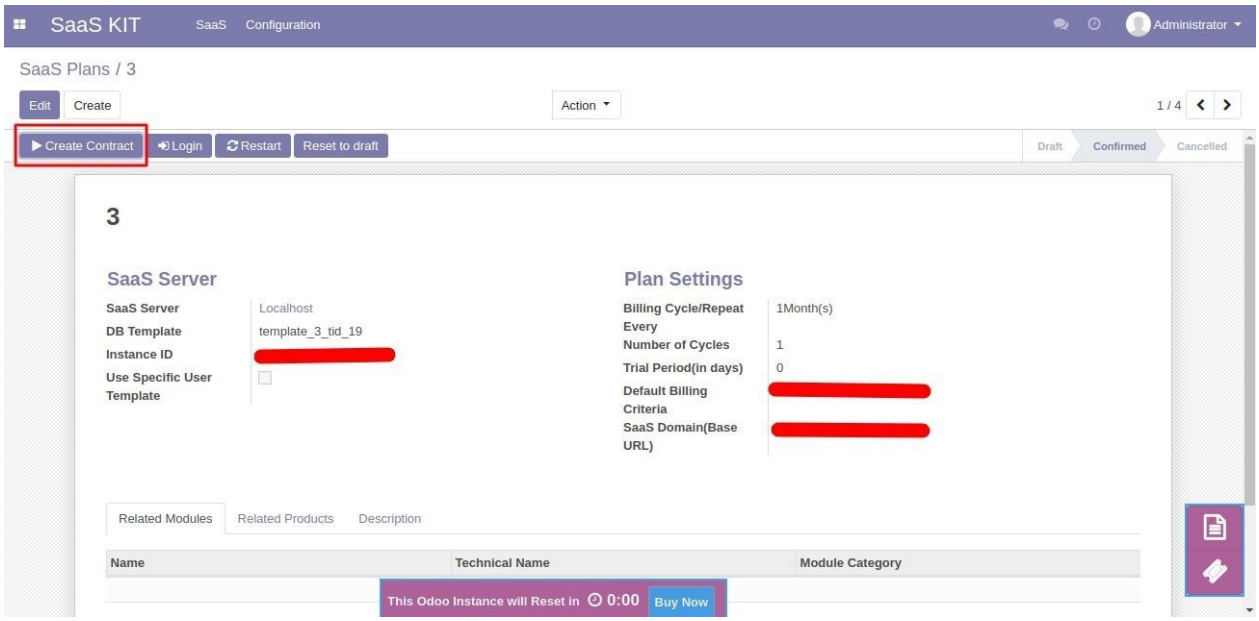
4.) Add **Related Modules** in the tab which you want to sell in this plan, only mentioned modules will be installed in client's instance.



5.) Now click on **Create DB Template** button. Tt will create the db template of this plan having all the modules installed which you have listed in **Related Modules**.

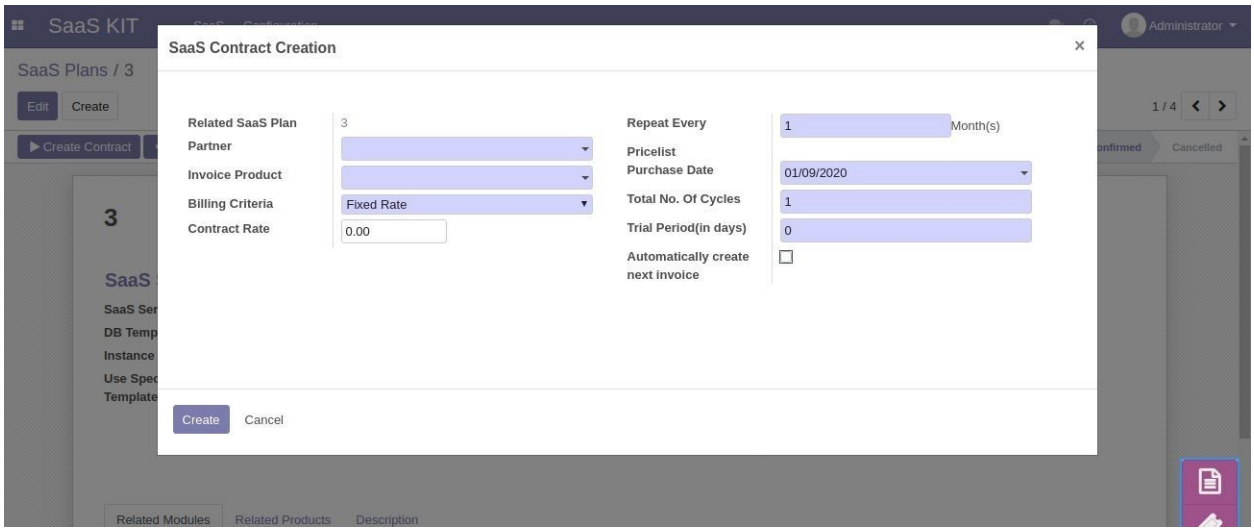


6.) After successful creation of db template. Click on **Create Contract.**

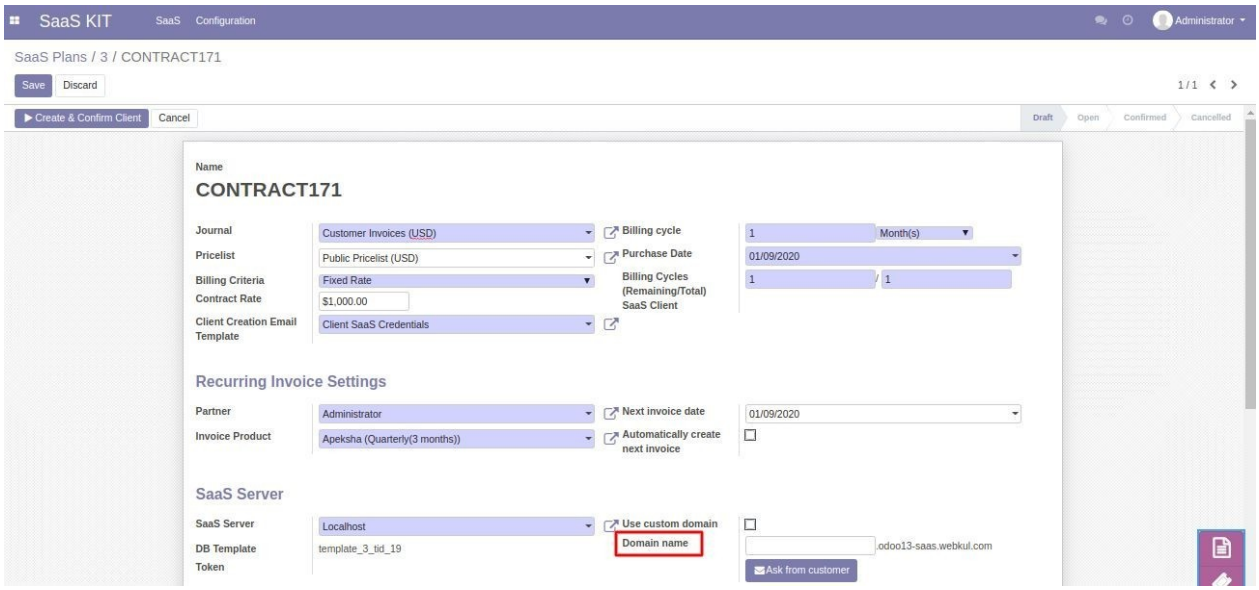


7.) On the pop up that comes up, mention the details required in the wizard for selling the plan.

8.) Select Partner Name to whom you want to sell this plan, Enter contract price for the plan then click on Create to create the contract of the plan for the selected partner, a contract will be created for the customer.

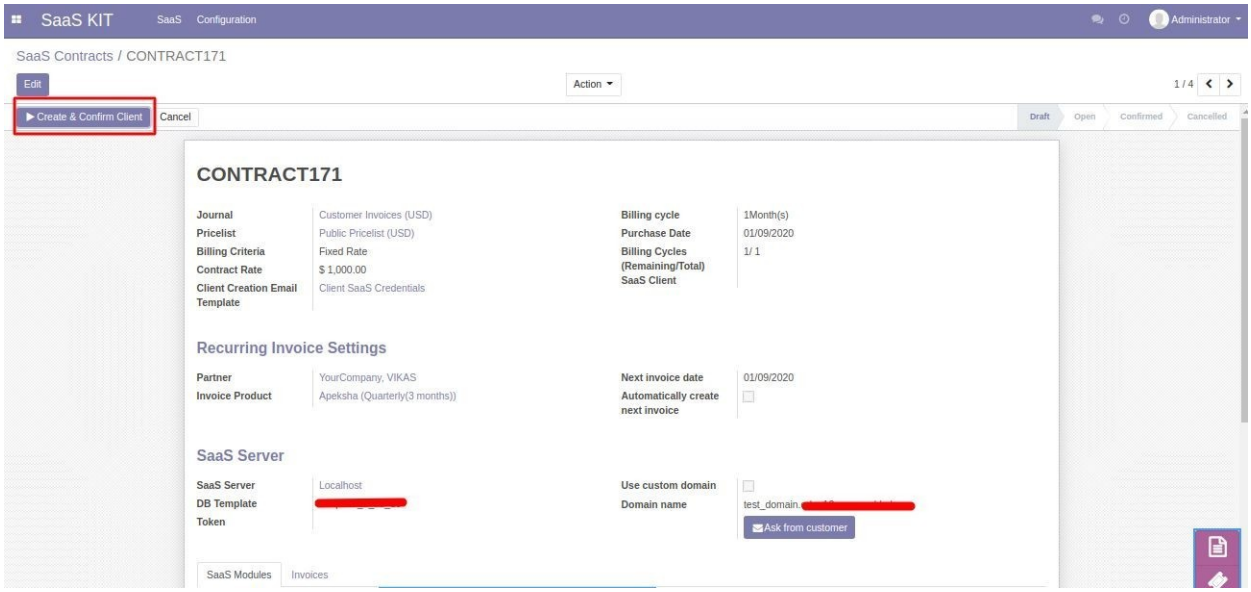


9.) Now, to create Odoo Saas instance for the particular customer enter the domain name on which customer's Odoo instance would run.



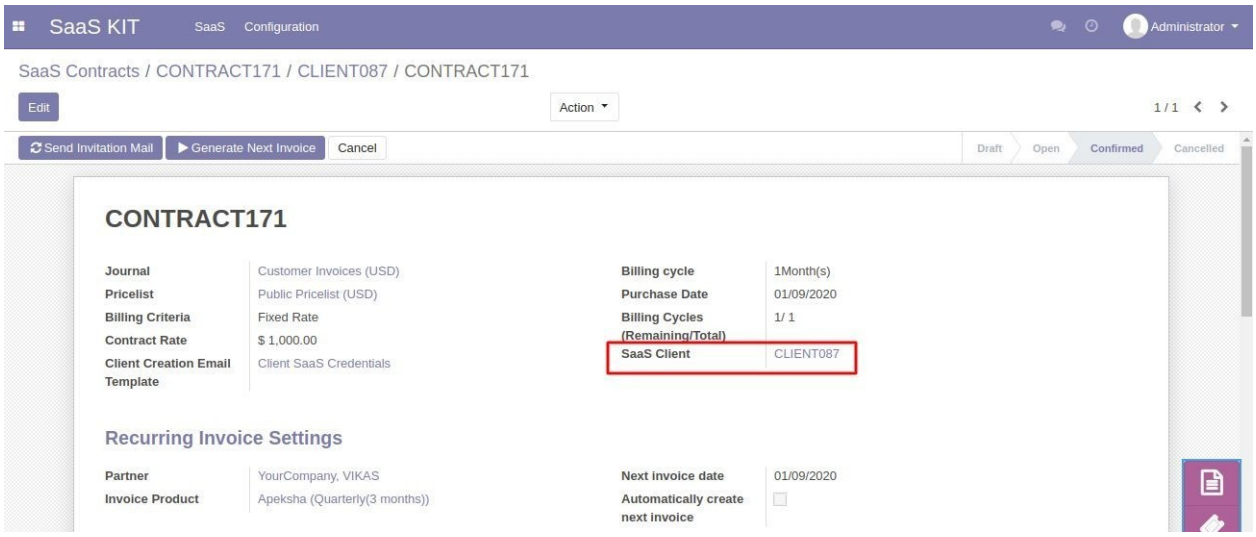
10.) For entering the domain name you can manually enter a unique domain name in Domain Name which have not used by any other customer till now.

or you can click on Ask from Customer button which will send a mail to the respective customer in which they will get a link to select a domain name of their choice, if customer entered a valid and a unique domain name then that domain will assign to his instance after auto creation of instance.

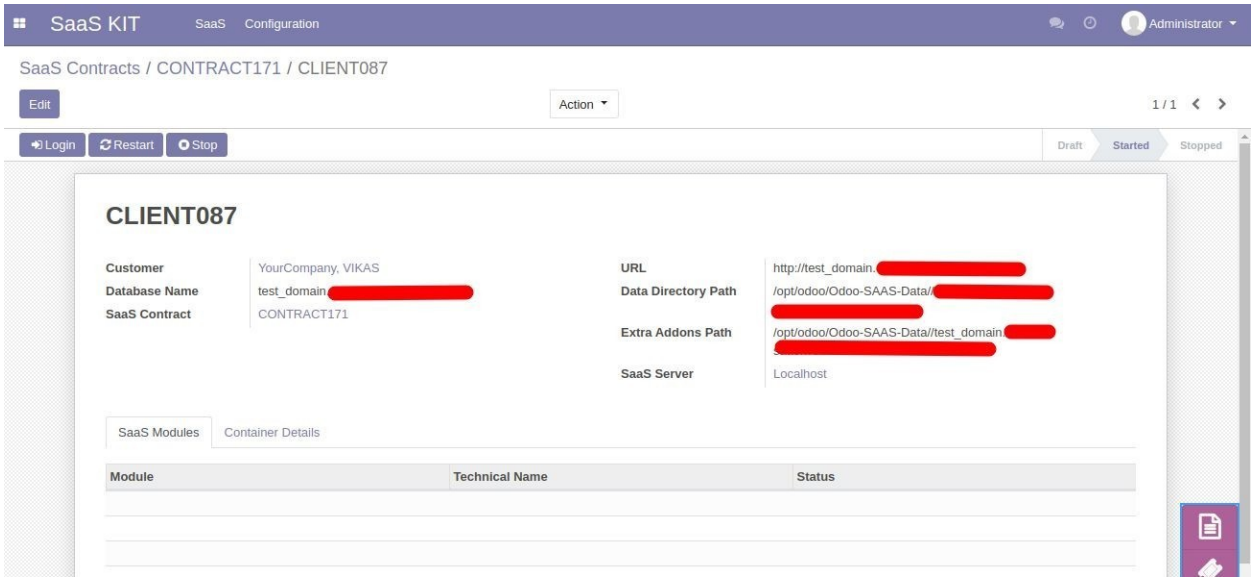


11.) In case of manually entering of domain name, click on save to save the details then click on **Create and Confirm Client** button which will create the odoo instance on selected domain for customers and create a client record, a mail will be sent to customer having the link to setup its password for odoo and the url to open his odoo.

12.) You can check the details of client's instance in **SaaS Client**.



13.) After click on SaaS client a client record will be shown having all the details of client.



14.) You can Start , Stop , Restart the customer's odoo instance by the help of buttons available on client's record.

15.) You can also login to the customer's instance by clicking in **Login** button on client's record.

Frequently Asked Questions

1.) Upload modules

There are two possible situations:-

i) Common Modules i.e modules that you want all SAAS clients to use:-

The modules should be placed in a common location which is reachable to all SAAS Clients & Templates. E.g the location is "/opt/odoo/common-addons" in your setup.

ii) SAAS Client specific Modules i.e modules that you want specific clients to use:-

The modules should be placed in location only reachable to the specific SAAS Client.

E.g:

The location is "/opt/odoo/Odoo-SAAS-Data/{client_container_name}/data-dir/addons/11.0"

Replace {client_container_name} with actual container name (client domain without http://. without braces).

Note:- After adding the module, the permissions/ownership should be set to Odoo user i.e "odoo" in your case.

2. Domain Directory Management

The individual virtual hosts file are kept in "/opt/odoo/Odoo-SAAS-Data/docker_vhosts" folder. One needs to have basic & sufficient knowledge of Nginx in order to make direct

changes in this folder otherwise refrain from doing so.

3. **Restart the server**

One can start/stop the SAAS client server from Odoo SAAS kit panel. The suggested method is to perform the operation from the server backend i.e. terminal on the server.

Use below command to restart any client container:-

docker restart {client_container_name}

Replace {client_container_name} with actual container name (client domain without http://. without braces).

I have queries:

1- Do I need to test, could I create clients and plans and then delete normally, or is there any way to test before working seriously with clients?

Reply:- We would you to suggest you to create a template & a corresponding client just to test & review the procedure & working.

Once confirmed you can delete the SAAS client & proceed ahead accordingly.

2- Where can I find the log file of the SAAS instance?

Reply:- You are expected to have basic understanding of Docker & its commands. You can check the logs of any client instance by running below command on your Server.

Command:-

docker exec -it {client_container_name} bash -c "tail -f /var/log/odoo/odoo-server.log"

Replace {client_container_name} with actual container name (client domain without http://. without braces).

3- How to backup and restore SAAS instance?:- Postgres/RDS backup, source code -- from image, filestore :- local script

Reply :- As the SAAS kit creates databases on the Postgresql Server or Managed Database Server(e.g. RDS). Thus, the backup depends on that.

Since, your saas has postgresql database hosted locally, you can backup the postgresql server accordingly.

4- What is the SAAS master password?:-

Reply:- Every client that gets created by Odoo SAAS Kit has its own Odoo configuration file e.g. /odoo/Odoo-SAAS-Data/{client_container_name}/odoo-

server.conf. That file contains master password for that client instance specifically.

Replace {client_container_name} with actual container name (client domain without http://. without braces).

Generally, the saas.conf i.e /odoo/webkul_addons/odoo_saas_kit/models/lib/saas.conf contains default master password. And every new client instance inherits its master password from this file.

5- How can I restart the process of specific client?

Reply:- You can restart the client instance from the Odoo SAAS Kit panel. Rather, we would recommend you to restart the client specific instance/container from the terminal of the main server. You can follow below command as per the need:-

Command :- **docker restart container_name**