**TEKO Odoo - Setup & Configure**

# Table of Contents

[Table of Contents 1](#_Toc18869)

[Setting Git by adding SSH key 2](#_Toc18962)

[Install Git for Windows 10 2](#_Toc22890)

[Generating and adding a SSH key to your profile 2](#_Toc27530)

[Install Linux and TEKO Odoo application 3](#_Toc24676)

[Install VirtualBox 3](#_Toc32023)

[Installing Vagrant 3](#_Toc32364)

[Understanding Vagrant 4](#_Toc22673)

[Setup a Vagrant machine with a VagrantFile 4](#_Toc16997)

[Run Vagrant using the VagrantFile 4](#_Toc9236)

[Run TEKO Odoo application from virtual machine 5](#_Toc18866)

[Configure TEKO Odoo 7](#_Toc16455)

[TEKO Odoo - Deploy on Ubuntu with GitlabCI (without using virtual machine) 7](#_Toc4076)

[Access virtual machine’s database from host machine 7](#_Toc2498)

[Appendix 9](#_Toc6434)

[Error when run vagrant up 9](#_Toc3784)

[Adding a SSH key to your profile 10](#_Toc13112)

[Error running the command: virtualenv --python=python3 .venv 12](#_Toc9135)

[Changing the database you are connnecting to 13](#_Toc12067)

# Setting Git by adding SSH key

## Install Git for Windows 10

- Download git from this link: <https://git-scm.com/download/win>

or link <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

- And then install it by default options.

## Generating and adding a SSH key to your profile

- Generating a new SSH key, see more steps here:

<https://git-scm.com/book/pl/v2/Git-on-the-Server-Generating-Your-SSH-Public-Key>

or

<https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/>

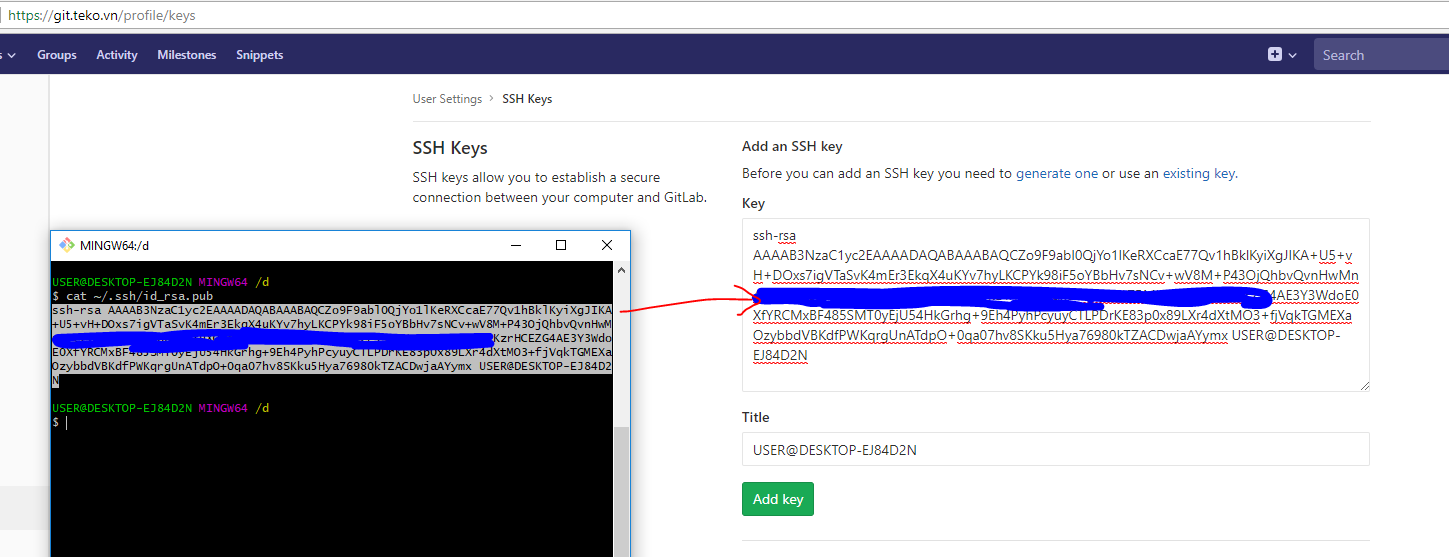
For example, run command and enter only for any prompts:

ssh-keygen -t rsa -b 4096 -C "*truc.nv@teko.vn*"

- Copy the content from **~/.ssh/id\_rsa.pub** file.

- Access to the GitLab website <https://git.teko.vn/profile/keys>

- Paste the copied content to Key box, enter a title and submit to add a new key



# Install Linux and TEKO Odoo application

## Install VirtualBox

Download VirtualBox for Windows and install it: <https://www.virtualbox.org/wiki/Downloads>

## Installing Vagrant

From <https://www.vagrantup.com/intro/getting-started/install.html>:

1. Installing the appropriate package for your system: <https://www.vagrantup.com/downloads.html>
2. Check that vagrant is installed by **opening a new command prompt or console**, and checking that vagrant is available:

$ vagrant

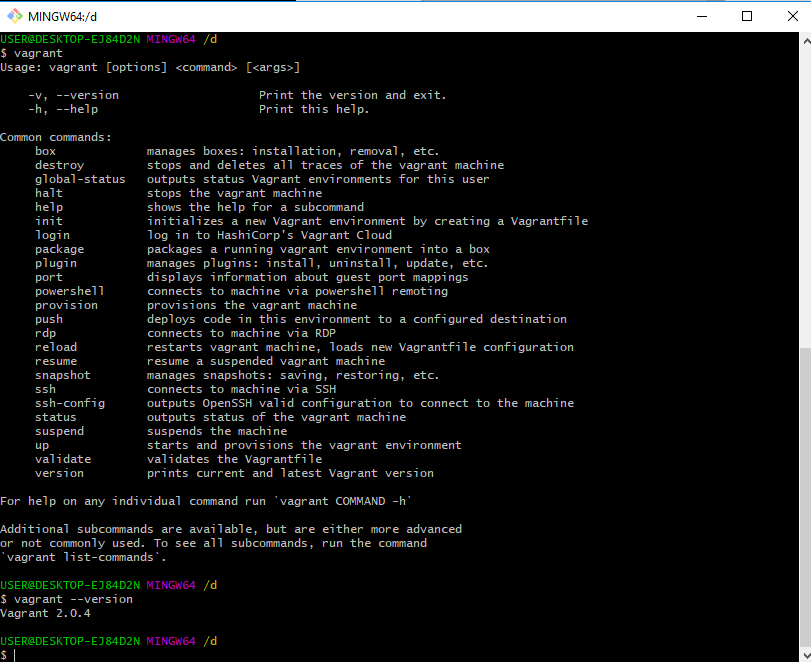
Usage: vagrant [options] <command> [<args>]

-v, --version Print the version and exit.

-h, --help Print this help.

# ...

*i.e. Output on a windows machine:*



## Understanding Vagrant

The links below were found to be useful:

<https://www.sitepoint.com/vagrantfile-explained-setting-provisioning-shell/>

<http://techie-notebook.blogspot.com/2015/10/understanding-vagrant-boxes-vms.html>

More CLI: <https://www.vagrantup.com/docs/cli/>

## **Setup a Vagrant machine with a VagrantFile**

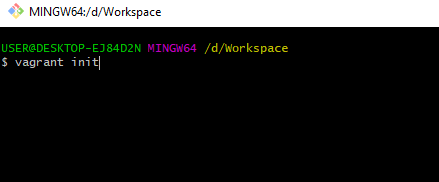
- Make a directory to store the Ubuntu virtual machine on which we will install Odoo. For example, **D:\workspace**

- Copy the **.ssh** directory from **C:\Users\%UserProfile%** of the Windows to the **D:\workspace\** location. This is for sharing the key pair on our host machine to be able to pull from a private repository on the virtual machine.

- Save the file **odoorc** (from the link: <https://drive.google.com/open?id=1xMIlAfIlaANoIhyDbMYkH1vWdTVFJ63A)> to the **D:\workspace\** location

- Open the “**Git Bash**” or **Windows Console Command Line**, point to the location **D:\workspace\** and run command line:

|  |
| --- |
| vagrant init |



- Download the **Odoo11CE\_VagrantFile\_update** (<https://drive.google.com/open?id=1ABKk0PmA1Ageua3a8sLomKR4e0PXP0tR>), then copy the contents of the file into the newly created VagrantFile in the **D:\workspace\** directory.

\* Let change highlighted text of the file to suit your information, as below:

- config.vm.network "private\_network", ip: "**192.168.102.18**"

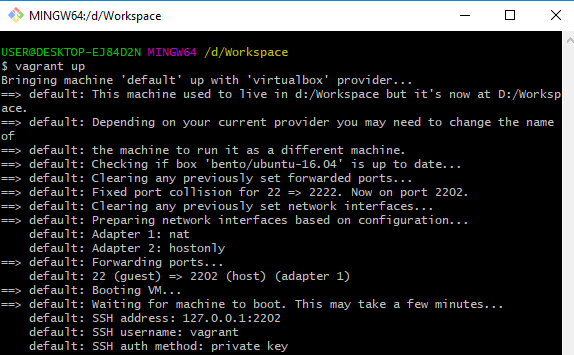
- config.vm.synced\_folder "**D:/Workspace**", "/workspace"

|  |
| --- |
| echo -e "Host git.teko.vn\n\tUser **truc.nv**\n\tStrictHostKeyChecking no\n\tIdentityFile ~/.ssh/id\_rsa" >> /home/vagrant/.ssh/config  echo -e "Host git.teko.vn\n\tUser **truc.nv**\n\tStrictHostKeyChecking no\n\tIdentityFile ~/.ssh/id\_rsa" >> /root/.ssh/config |

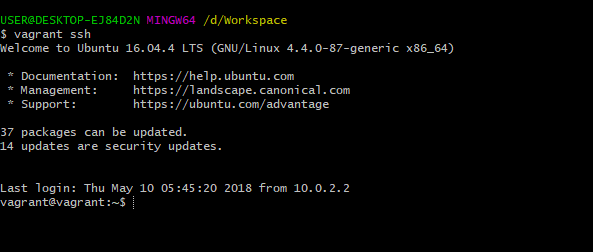
## Run Vagrant using the VagrantFile

- **Prerequisites**: must have permission to access TEKO’s git. If you do not have it, let ask your leader sets the permission.

- Run the console command: **vagrant up** to creates and configures guest machines according to the Vagrantfile. It will make a Linux OS and setup Odoo project on the OS

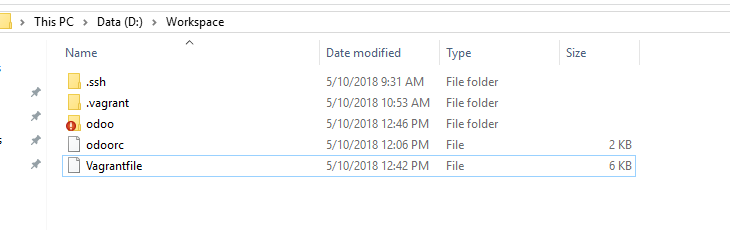


- Run the command: **vagrant ssh** to access the vagrant machine

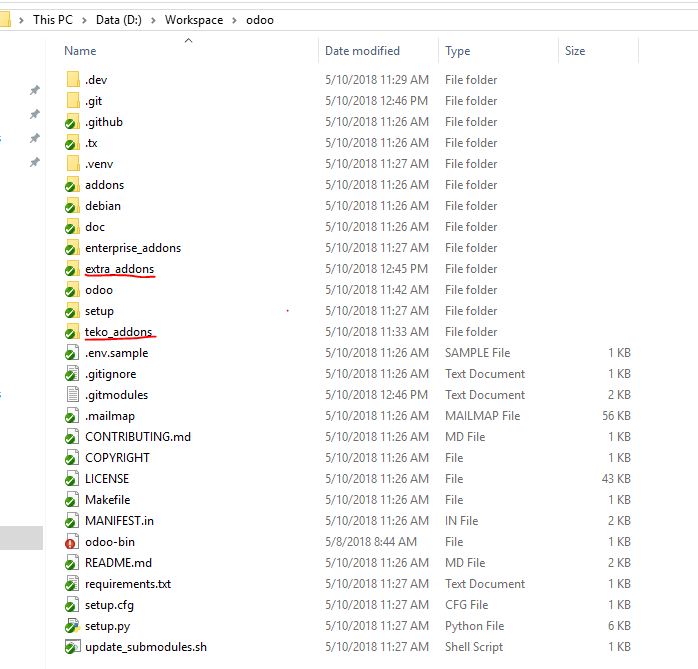


# Run TEKO Odoo application from virtual machine

- Check source from D:\Workspace after run command: vagrant up

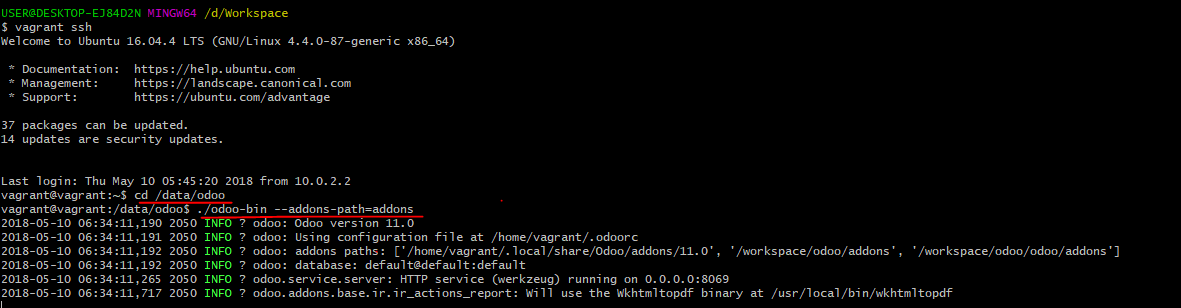


\* It added the .vagrant hidden folder and odoo folder with contents:

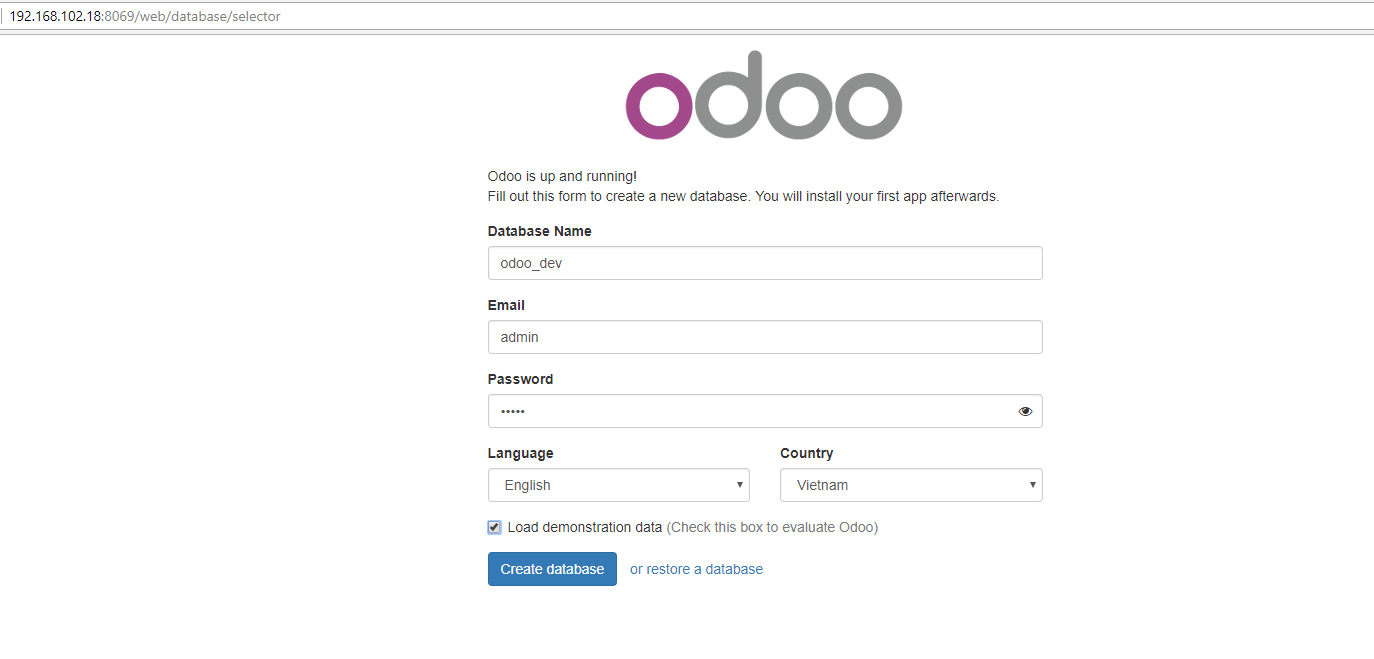


- On the command line, after running the command vagrant ssh, you can start odoo by using the frequently commands:

|  |
| --- |
| cd /data/odoo  ./odoo-bin addons\_path = /data/odoo/addons,/data/odoo/extra\_addons,/data/odoo/enterprise\_addons,/data/odoo/teko\_addons  #or using command below:  ./odoo-bin -c ~/.odoorc |



And then run the browser with this link: <http://192.168.102.18:8069> to init information (database, login account) of the Odoo application as below capture:



# Configure TEKO Odoo

Check details from link below: <https://docs.google.com/document/d/1eUw0GNiooC0L_dVg_yQH_xzVyJWVYdGerxO_1Nx2l40/edit>

# TEKO Odoo - Deploy on Ubuntu with GitlabCI (without using virtual machine)

After coding to customize the Odoo app, we can deploy the application on a Linux server by using GitlabCI that without using a virtual machine. Read more details from this link:

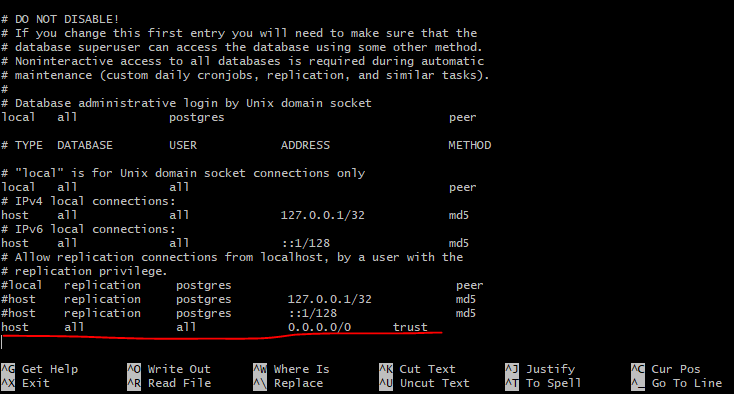
<https://docs.google.com/document/d/1UOU21agivyKv4WQst0okpZnRvgZmVOuvlFBBZRUDTQs/edit>

# Access virtual machine’s database from host machine

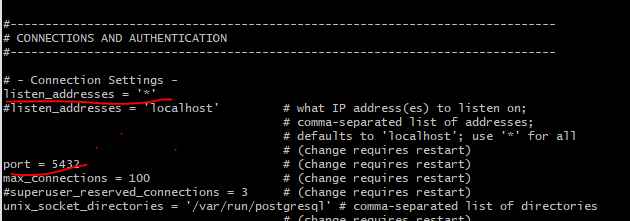
Access to the virtual machine as ‘vagrant’ user and then configure Postgres that use frequently steps below:

- Update **pg\_hba.conf** (most likely in **/etc/postgresql/9.4/main**) with -

host all all 0.0.0.0/0 trust

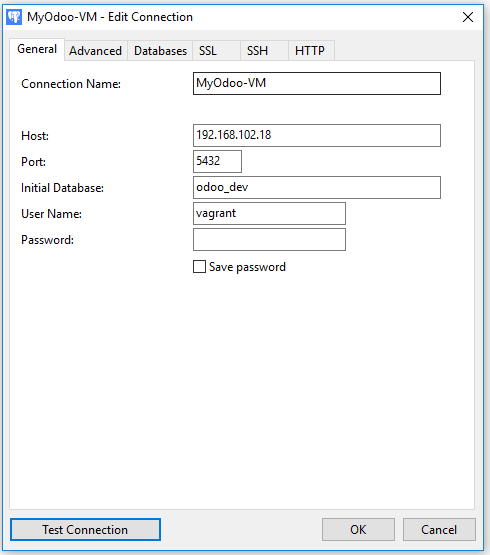


- Update **postgresql.conf** to use **listen\_addresses = '\*'**



- Be sure to **sudo service postgresql restart**

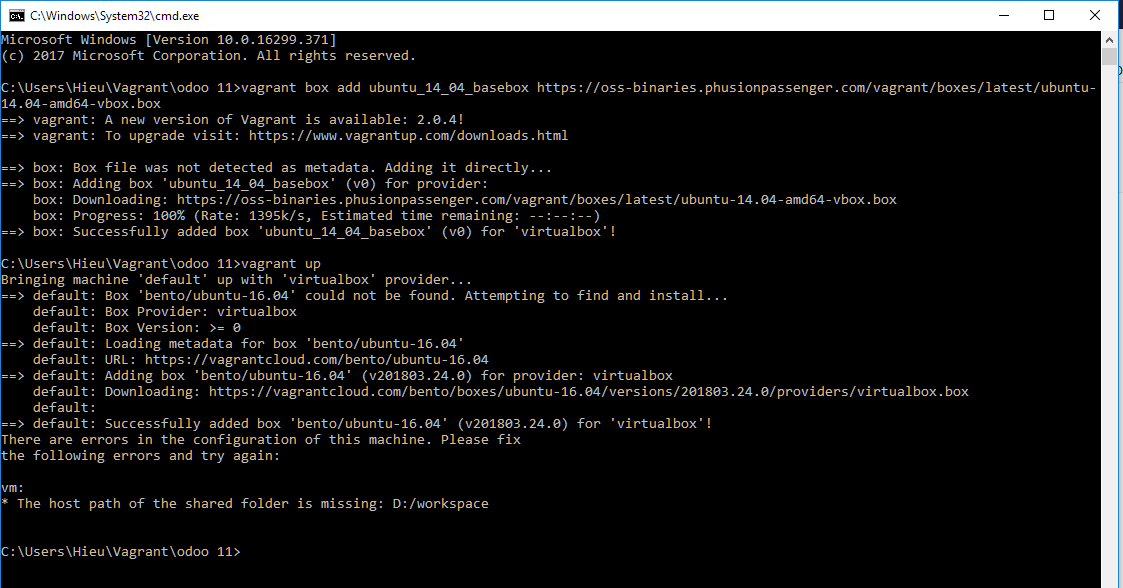
- Test to connect the database from host machine (using DB Navicat 12 download from link <https://1drv.ms/f/s!AttyPZdhUIEnzCyzjfWgU1_2AEWf>)



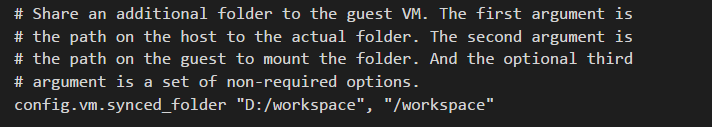
# Appendix

## Error when run vagrant up

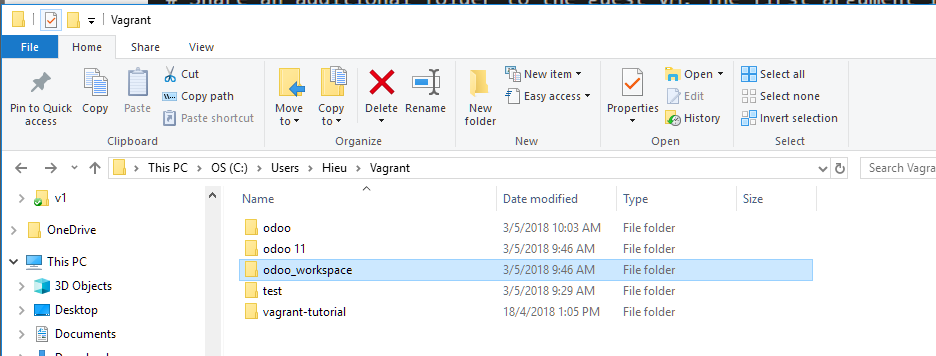
You may encounter the following error:



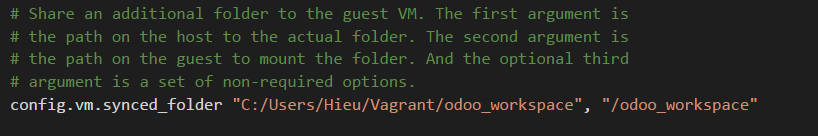
This is because of the following line in the VagrantFile:



The folder probably does not exist on your machine. All you need to do is create a new workspace folder and then change the path of the synced folder. To illustrate, I created a new **Odoo\_workspace** folder in my Vagrant folder:

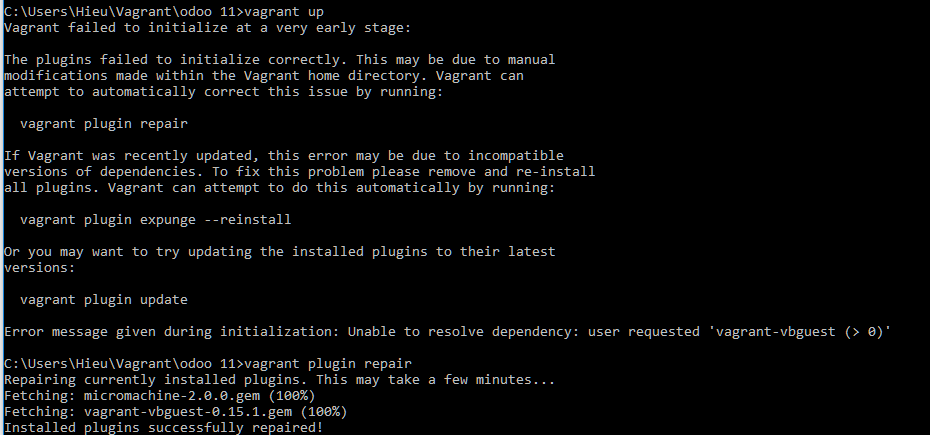


and changed the arguments passed to config.vm.synced\_folder to:

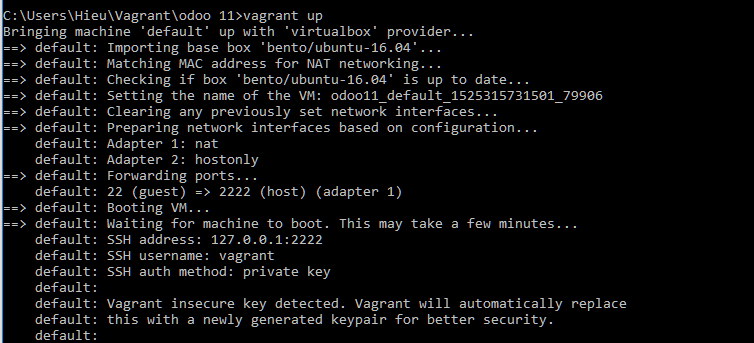


Run the console command: **vagrant plugin repair**

(The below snapshot illustrates the fact that running vagrant up will not work because vagrant failed to initialize the machine the first time booting it up)



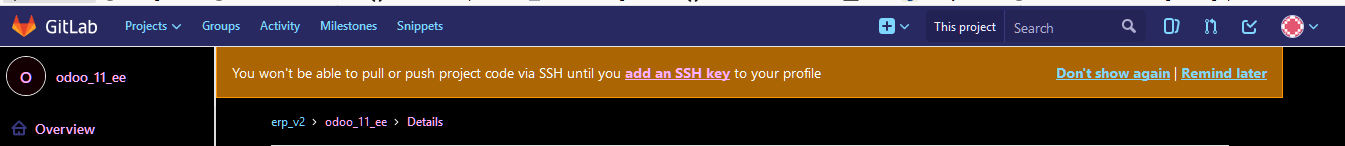
Run the command : **vagrant up** again.

 Note: It may take a while as the necessary libraries and components are installed and configured!

The Vagrant machine is now up and running with the required components installed!

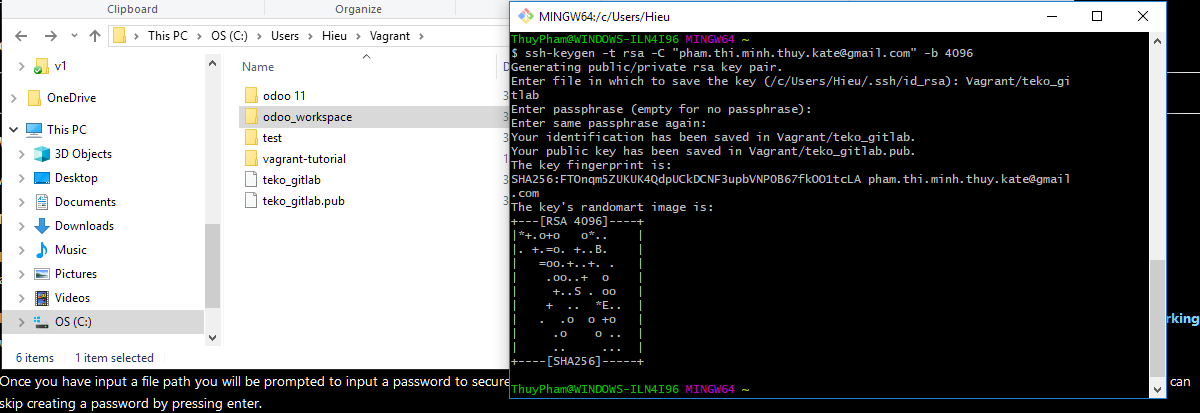
## Adding a SSH key to your profile

If you see the following popup, continue reading else skip this section.



You will need to have installed Git on your windows machine; Refer to the following link if you not yet installed Git: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

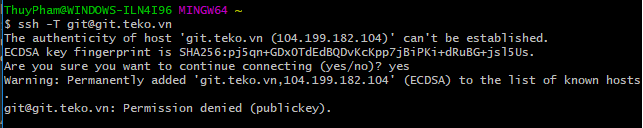
Click on the “add an SSH key” hyperlink in the popup. Follow the instructions. Since I could not locate an existing key pair, I had to generate one like such (using GitBash):



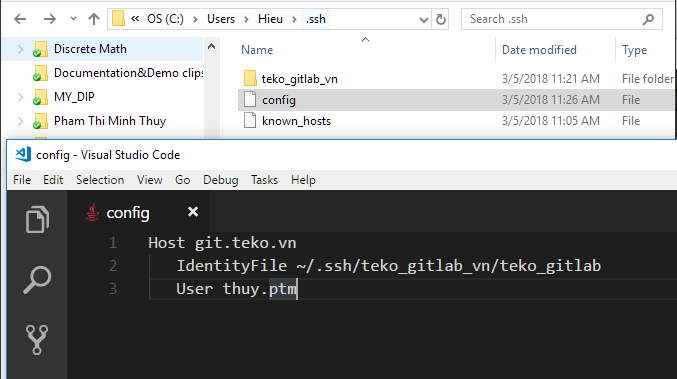
After adding the SSH public key to your profile, the popup should no longer appear.

Take note of the location you saved the SSH key pair in.You will need this later on.

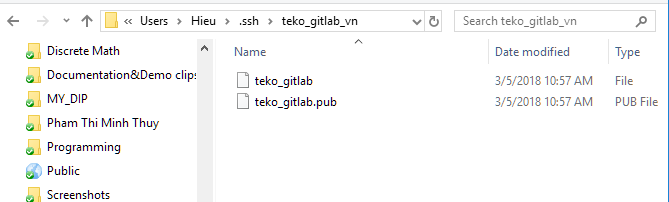
Now we create a config file in the ./ssh folder (full file path is C:\Users\%UserProfile%\.ssh by default ) so that we do not get the following error when testing our access to git@git.teko.vn:



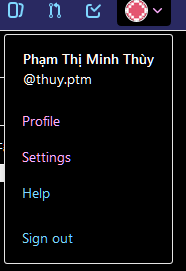
The config file has no extension. I chose to store it in my ./ssh folder:



Change the path of the IdentityFile variable to the location of the your private Key. Note that ~/ is a shortcut for C:\Users\%UserProfile% :



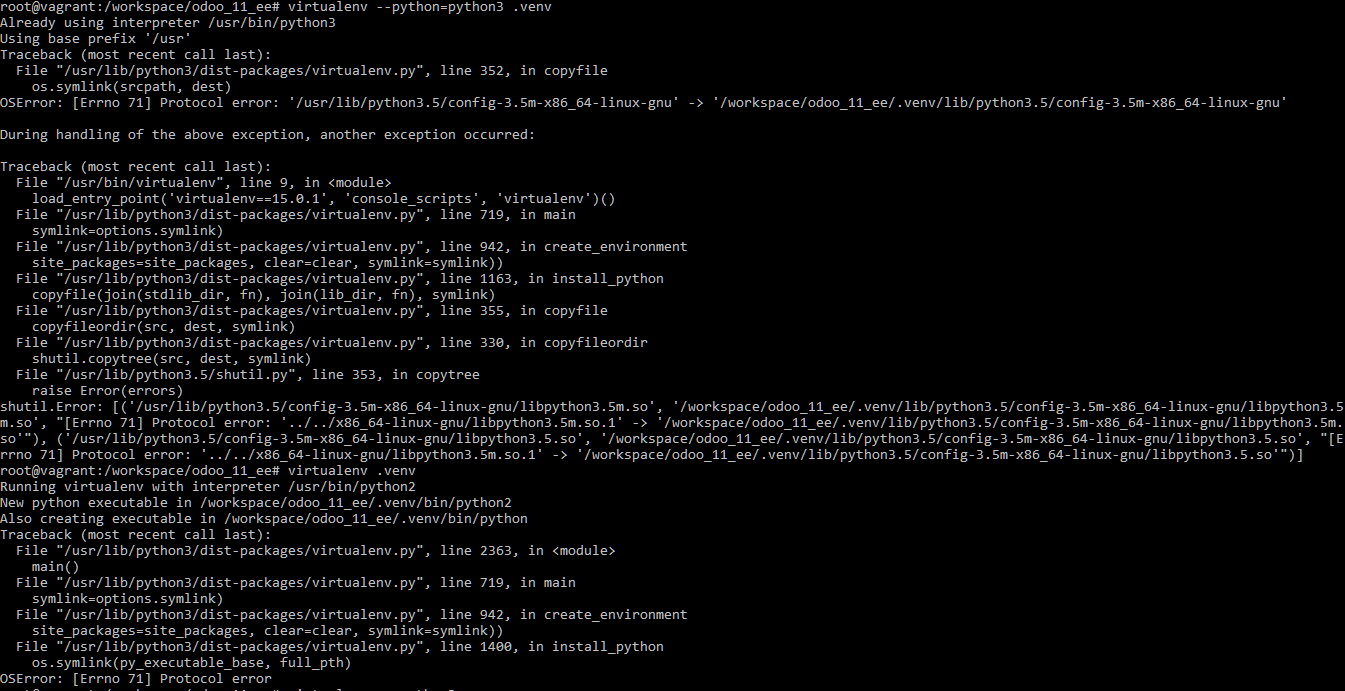
Change the User variable value. Your value = gitlab username can be found here:

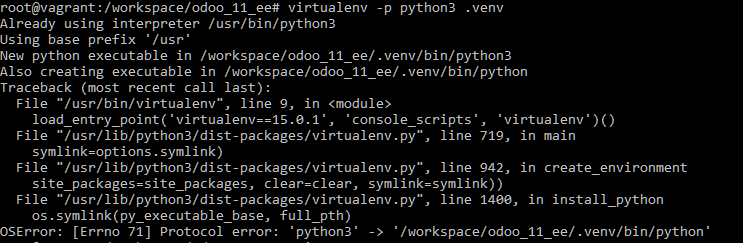


Running the command in GitBash should now work: 

Able to successfully access a private repository using SSH on the host machine.

**Error running the command: virtualenv --python=python3 .venv** (in your odoo project folder)

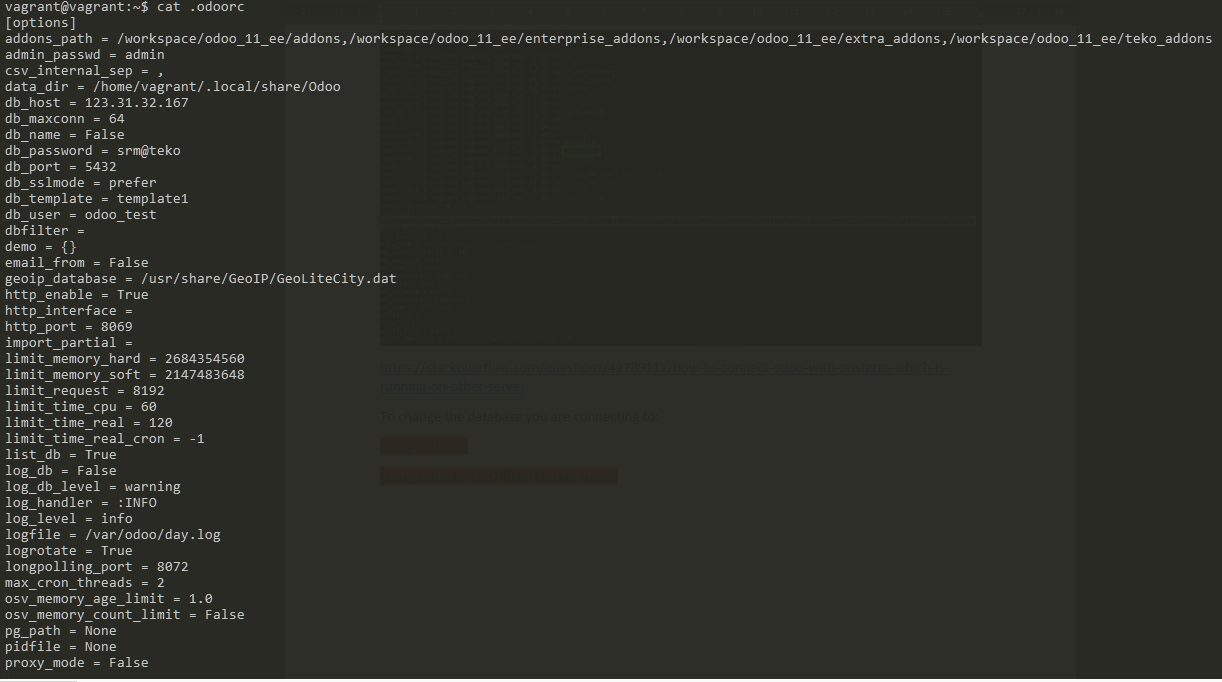


Running an equivalent command to create a virtualenv: virtualenv -p python3 .venv also produces an error.

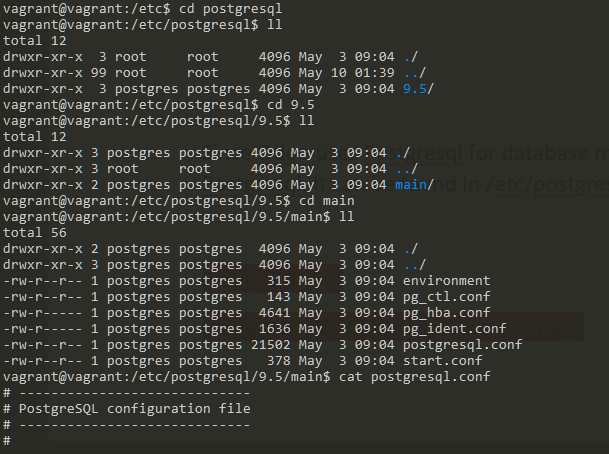
Fix: Just start your git bash/cmd prompt console as an administrator. Then, vagrant up > setup your virtual env(<https://stackoverflow.com/questions/24640819/protocol-error-setting-up-virtualenvironment-through-vagrant-on-ubuntu>

**Changing the database you are connnecting to**

Your Odoo configuration file is .odoorc. To change the database you are connecting to, you need to modify this file:

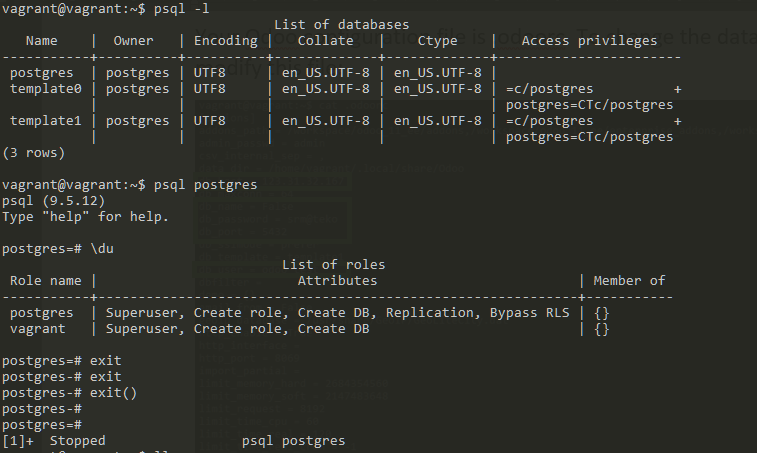


Since Odoo uses Postgresql for database management, we need to look at the configuration file for Postgresql. It can be found in /etc/postgresql/%version\_number%/main/postgresql.conf

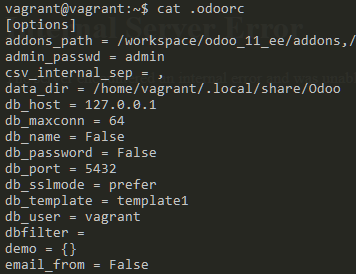


To list the databases currently in postgres, use the command : **psql –l**

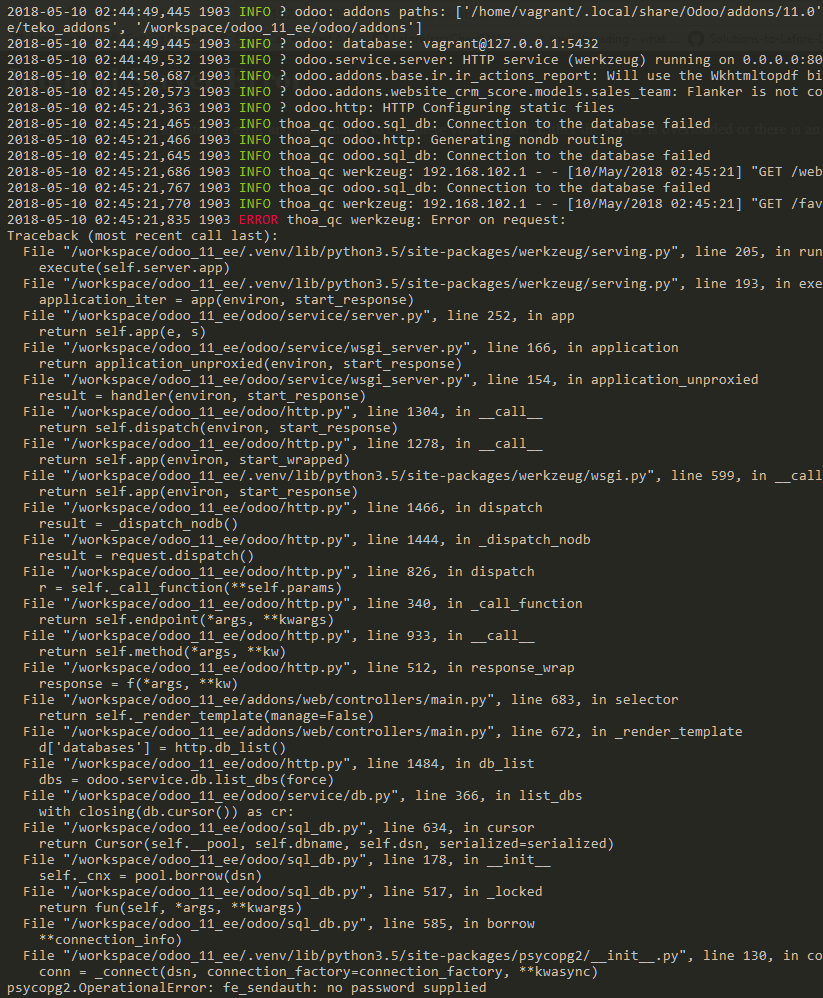
To access the database, use the command **psql postgres** ; to exit the database, use the command: **Ctrl - Z**

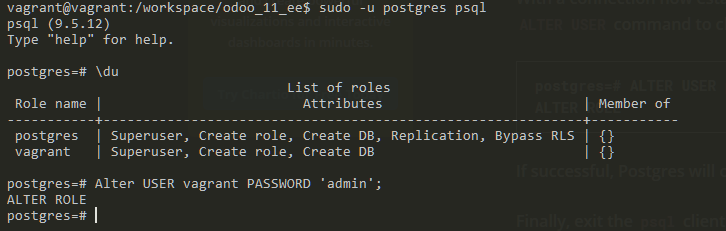


Modify the .odoorc file database settings using the database information provided in postgresql.conf:

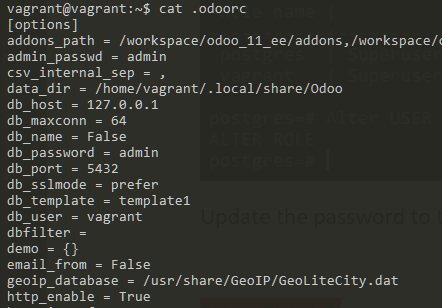


I got the following error when I ran the Odoo server:





Update the password to the one that has just been set:



Rerun ./odoo-bin, you should be greeted by the following when accessing the website:

