# Rails-app-in-a-Vagrant-box

**Author:** Sudip Prajapati

**Date:** 18/10/2021

Configuration as well as hosting of the rails app in a Vagrant box which is simplified with some process down to the following steps:

* Installation of Virtual Box (Oracle or can be any other virtual machine)
* Installation of Hashicorp Vagrant
* Creation of local directory for Vagrant
* Run Vagrant up and configuration of the Vagrant box
* Usage of Hashicorp/bionic64 vagrant box to continue the given process
* Run the rails server which should be accessible from your local browser

**Starting of Hashicorp Vagrant**

* Vagrant init and Vagrant up is done with the continuation of the respective command as per required in the vagrant meeting the running procedure of ruby and postgresql database as well as rvm installation to further the configuration.

**Requirements:**

* Installation of RVM
* Ruby must be of version -2.4.1
* Postgresql database
* Rails
* The system must be updated i.e. sudo apt-get update
* Gem file to run the application i.e. we can get it form bundle install
* Curl = \curl -sSL <https://get.rvm.io> | bash -s stable
* Gpg2 = sudo apt-get install gnugp2

**Obstacles faced during the configuration process**

* As per the requirements given above, the ruby version must be of 2.4.1 and the first obstacle or we can say as the problem faced is the vagrant system doesnot recognize the required version for the configuration process which should be kept noted and installed the required version i.e. 2.4.1 manually with the respective command required.
* The deletion of the Gemfile.lock is required before running of the bundle install which helps to avoid the not supporting problem in the project as the gemfile.lock is aready contained with the package name.

[Note: Gemfile.lock already contains package name which doesnot support in the respective project.]

* Gemfile should be installed manually when doing bundle install and we should make sure that gem install pg -v '0.21.0' --source 'https://rubygems.org/' succeeds before bundling.
* After the hosting of simple web using rails server the username not authenticated was displayed which was solved by creating the username as well as the database in the postgres with peer authentication problem by changing the authentication to md5 by peer and the restarting of postgres is must.

• When doing bundle install in project directory make sure first delete/remove Gemfile.lock and then run bundle install. [because gemfile.lock already contains package name which didn’t support in your project].

• Some of the gem file didn’t install when doing bundle install so, we have to install them manually like, [An error occurred while installing pg (0.21.0), and Bundler cannot continue. Make sure that gem install pg -v '0.21.0' --source 'https://rubygems.org/' succeeds before bundling.] In this case, we have to install manually i.e. gem install pg -v '0.21.0' --source '<https://rubygems.org/>'

After all the issues were resolved finally, we were moving head towards creating database. As we are using postgesql I have faced so many problems during setup database.

• First when creating role with password for Postgresql database, su - postgres [error: No passwd entry for user 'postgres']

After some googling I found the solution here and needs to restart postgres service sudo service postgresql restart

• Then I need to give the admin privilege to user ‘postgres’ for the respective database.

• Then edit database.yml file, define your own database and user pwd here, for eg please see the attached image.

Note: Make sure you update your repository list.

OR,

Enable the PostgreSQL apt repository

Create the file “/etc/apt/sources.list.d/pgdg.list” and add the corresponding line for the repository in it: vagrant@vagrant:# sudo sh -c 'echo "deb <http://apt.postgresql.org/pub/repos/apt/> bionic-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'

Import the repository signing key and update the package lists: vagrant@vagrant:~# wget --quiet -O - <https://www.postgresql.org/media/keys/ACCC4CF8.asc> | sudo apt-key add –

Then update your repository list i.e., sudo apt-get update

Boom :D you can connect to PostgreSQL

Now you can login to your cluster vagrant@vagrant:~# ps -ef | grep postgres

The most important part is make sure you forward port in vagrantfile because to run the hosted app in rails server you have to run from host machine.

When done you can run your rails server, rails s -b domainname

Finally, a simple web is hosted using rails server

\*\*YOU CAN FIND DEMO\_IMG IN \*\*

Rails-app-in-a-Vagrant-box/demogif/