# Vagrant Virtual Machine Spin up

Tools used:

1. Vagrant: Version 1.9.1
2. Virtual box: Version 5.1.14
3. CentOS: Version 7

Spinning up instance using the below Vagrant file

|  |
| --- |
| # -\*- mode: ruby -\*- |
|  | # vi: set ft=ruby : |
|  |  |
|  | # All Vagrant configuration is done below. The "2" in Vagrant.configure |
|  | # configures the configuration version (we support older styles for |
|  | # backwards compatibility). Please don't change it unless you know what |
|  | # you're doing. |
|  | Vagrant.configure(2) do |config| |
|  | # The most common configuration options are documented and commented below. |
|  | # For a complete reference, please see the online documentation at |
|  | # https://docs.vagrantup.com. |
|  |  |
|  | # Every Vagrant development environment requires a box. You can search for |
|  | # boxes at https://atlas.hashicorp.com/search. |
|  | config.vm.box = "centos/7" |
|  |  |
|  | # Disable automatic box update checking. If you disable this, then |
|  | # boxes will only be checked for updates when the user runs |
|  | # `vagrant box outdated`. This is not recommended. |
|  | # config.vm.box\_check\_update = false |
|  |  |
|  | # Create a forwarded port mapping which allows access to a specific port |
|  | # within the machine from a port on the host machine. In the example below, |
|  | # accessing "localhost:8080" will access port 80 on the guest machine. |
|  | config.vm.network "forwarded\_port", guest: 80, host: 8080 |
|  |  |
|  | # Create a private network, which allows host-only access to the machine |
|  | # using a specific IP. |
|  | # config.vm.network "private\_network", ip: "192.168.33.10" |
|  |  |
|  | # Create a public network, which generally matched to bridged network. |
|  | # Bridged networks make the machine appear as another physical device on |
|  | # your network. |
|  | # config.vm.network "public\_network" |
|  |  |
|  | # Share an additional folder to the guest VM. The first argument is |
|  | # the path on the host to the actual folder. The second argument is |
|  | # the path on the guest to mount the folder. And the optional third |
|  | # argument is a set of non-required options. |
|  | # config.vm.synced\_folder "../data", "/vagrant\_data" |
|  |  |
|  | # Provider-specific configuration so you can fine-tune various |
|  | # backing providers for Vagrant. These expose provider-specific options. |
|  | # Example for VirtualBox: |
|  | # |
|  | # config.vm.provider "virtualbox" do |vb| |
|  | # # Display the VirtualBox GUI when booting the machine |
|  | # vb.gui = true |
|  | # |
|  | # # Customize the amount of memory on the VM: |
|  | # vb.memory = "1024" |
|  | # end |
|  | # |
|  | # View the documentation for the provider you are using for more |
|  | # information on available options. |
|  |  |
|  | # Define a Vagrant Push strategy for pushing to Atlas. Other push strategies |
|  | # such as FTP and Heroku are also available. See the documentation at |
|  | # https://docs.vagrantup.com/v2/push/atlas.html for more information. |
|  | # config.push.define "atlas" do |push| |
|  | # push.app = "YOUR\_ATLAS\_USERNAME/YOUR\_APPLICATION\_NAME" |
|  | # end |
|  |  |
|  | # Enable provisioning with a shell script. Additional provisioners such as |
|  | # Puppet, Chef, Ansible, Salt, and Docker are also available. Please see the |
|  | # documentation for more information about their specific syntax and use. |
|  | # config.vm.provision "shell", inline: <<-SHELL |
|  | # sudo apt-get update |
|  | # sudo apt-get install -y apache2 |
|  | # SHELL |
|  | config.vm.provision :shell, :path => "bootstrap.sh" |
|  | End |

And then I hit **C:\Users\phane\vagrantdemo> vagrant up** enter.

**Spinning up single CentOS instance.**

Tried spinning up centos7 using vagrant.

Created a folder named vagrant in D drive

Created a subfolder named centos, and ran vagrant init using git bash cli

Opened vagrantfile in the centos folder, removed all comments in the vagrant file saved and closed the file in CLI

Now ran $ vagrant up

Failed due to rsync error. Browsed internet for solution and found the below solution

config.vm.synced\_folder ".", "/vagrant", type: "virtualbox", **disabled: true**

after adding this line, I saved the document and entered vagrant up in cli and monitored VMbox. I can see centos up and running.

Tried ssh to the centos it successfully worked.

Task completed

**Create a Vagrant file to run a CentOS machine with Apache and Jenkins installed in it**

Created a directory **Apache Jenkins** to install a server with apache and Jenkins

Initiated Git init and vagrant init

**Github:** Created repository in github vapache\_jenkins

Added vagrantfile using $ git add vagrantfile

Checked the status: $ git status

once the status is good I moved to towards committing the change

$ git commit -m "Apache and Jenkins before setup start"

Then I have moved to pushing the committed content to a remote location “github” to do that 1st I need to run a command in the gitbash cli

$ git remote add origin <https://github.com/phaneindranath/vapache_jenkins.git>

Next pushing the committed changes in git

$ git push -u origin master

Result: Counting objects: 3, done.

Delta compression using up to 8 threads.

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 1.53 KiB | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/phaneindranath/vapache\_jenkins.git

\* [new branch] master -> master

Branch master set up to track remote branch master from origin.

Apache and Jenkins build