**Using Docker as provider for Vagrant**

[#linux](https://dev.to/t/linux)[#docker](https://dev.to/t/docker)[#tutorial](https://dev.to/t/tutorial)

According to the [documentation](https://www.vagrantup.com/docs/providers), Vagrant has support for [VirtualBox](https://www.virtualbox.org/), [VMWare](https://www.vmware.com/), [Hyper-V](https://www.microsoft.com/hyper-v) and [Docker](https://www.docker.io/) as providers.

Through this blog post you will learn how to configure Docker and Vagrant.

**Docker**

For installing Docker on Linux follow the instructions in the [documentation](https://docs.docker.com/engine/install/). Go directly to the instructions of your distribution from the following list:

* [CentOS](https://docs.docker.com/engine/install/centos)
* [Debian](https://docs.docker.com/engine/install/debian)
* [Fedora](https://docs.docker.com/engine/install/fedora)
* [Raspbian](https://docs.docker.com/engine/install/debian)
* [Ubuntu](https://docs.docker.com/engine/install/ubuntu)

If you use Arch Linux or any Arch-based distribution, install it using pacman and initialize the daemon:

$ sudo pacman -S docker

$ sudo systemctl start docker

For running Docker commands you will need root permissions. As Vagrant will run it on your behalf, you have to configure it for running without sudo. You can follow the instructions in the [Post-installation steps for Linux](https://docs.docker.com/engine/install/linux-postinstall/) section as described below.

First create the docker group:

$ sudo groupadd docker

Add your user to the docker group:

$ sudo usermod -aG docker $USER

You will have to log out and log back in for the changes to take effect.

If you want to activate the changes to groups in your current session, run:

$ newgrp docker

To check if you can run docker commands without sudo:

$ docker run hello-world

This command will download the test container hello-world and run it.

**Vagrant**

For installing Vagrant go to the [download page](https://www.vagrantup.com/downloads) and get the right package for your distribution. You can also install it from the repositories of some Linux distributions.

**Debian-based:**

$ sudo apt install vagrant

**Fedora:**

$ sudo dnf install vagrant

**CentOS**

$ sudo dnf install -y https://releases.hashicorp.com/vagrant/2.2.9/vagrant\_2.2.9\_x86\_64.rpm

**Arch Linux:**

$ sudo pacman -S vagrant

**Vagrant + Docker**

There are two ways you can use Docker as provider. Using an image from the Docker registry:

Vagrant.configure("2") do |config|

config.vm.provider "docker" do |d|

d.image = "foo/bar"

end

end

Or a Dockerfile:

Vagrant.configure("2") do |config|

config.vm.provider "docker" do |d|

d.build\_dir = "."

end

end

**Using a Dockerfile**

First you have to create a directory to store the configuration files for your environment and change to this directory.

$ mkdir docker-test

$ cd docker-test

Create a Dockerfile:

$ touch Dockerfile

And add the following content:

FROM ubuntu

ENV TZ=America/Mexico\_City

RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

RUN apt-get update -y

RUN apt-get install -y --no-install-recommends ssh sudo

RUN useradd --create-home -s /bin/bash vagrant

RUN echo -n 'vagrant:vagrant' | chpasswd

RUN echo 'vagrant ALL = NOPASSWD: ALL' > /etc/sudoers.d/vagrant

RUN chmod 440 /etc/sudoers.d/vagrant

RUN mkdir -p /home/vagrant/.ssh

RUN chmod 700 /home/vagrant/.ssh

RUN echo "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NF8iallvQVp22WDkTkyrtvp9eWW6A8YVr+kz4TjGYe7gHzIw+niNltGEFHzD8+v1I2YJ6oXevct1YeS0o9HZyN1Q9qgCgzUFtdOKLv6IedplqoPkcmF0aYet2PkEDo3MlTBckFXPITAMzF8dJSIFo9D8HfdOV0IAdx4O7PtixWKn5y2hMNG0zQPyUecp4pzC6kivAIhyfHilFR61RGL+GPXQ2MWZWFYbAGjyiYJnAmCP3NOTd0jMZEnDkbUvxhMmBYSdETk1rRgm+R4LOzFUGaHqHDLKLX+FIPKcF96hrucXzcWyLbIbEgE98OHlnVYCzRdK8jlqm8tehUc9c9WhQ==" > /home/vagrant/.ssh/authorized\_keys

RUN chmod 600 /home/vagrant/.ssh/authorized\_keys

RUN chown -R vagrant:vagrant /home/vagrant/.ssh

RUN sed -i -e 's/Defaults.\*requiretty/#&/' /etc/sudoers

RUN sed -i -e 's/\(UsePAM \)yes/\1 no/' /etc/ssh/sshd\_config

RUN mkdir /var/run/sshd

RUN apt-get -y install openssh-client

EXPOSE 22

CMD ["/usr/sbin/sshd", "-D"]

The official Docker image of Ubuntu will be used as specified in FROM ubuntu.

When running apt-get update -y or apt update -y, it will ask you to configure the timezone, the prompt will wait for you to enter the selected option.

To avoid this, you have to add the configuration options in the Dockerfile, as described [here](https://dev.to/setevoy/docker-configure-tzdata-and-timezone-during-build-20bk), by adding the following lines:

ENV TZ=America/Mexico\_City

RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

Replacing the value of TZ according to your timezone.

Vagrant requires an SSH connection to access the container and Docker images come only with the root user. You have to configure another user with root permissions. That's why the ssh and sudo packages are required.

In the following lines the vagrant user is created and a password assigned. The user wouldn't be required to use a password when running any command that requires root permissions. The user is also added to the sudo group.

RUN useradd --create-home -s /bin/bash vagrant

RUN echo -n 'vagrant:vagrant' | chpasswd

RUN echo 'vagrant ALL = NOPASSWD: ALL' > /etc/sudoers.d/vagrant

.ssh directory must be created. This is the directory when configuration files related with SSH connection are stored.

RUN mkdir -p /home/vagrant/.ssh

RUN chmod 700 /home/vagrant/.ssh

An insecure key is added for the initial configuration. This key will be replaced later when you initialize your virtual environment the first time. Also, the ownership of the .ssh directory is changed to vagrant user.

RUN echo "ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NF8iallvQVp22WDkTkyrtvp9eWW6A8YVr+kz4TjGYe7gHzIw+niNltGEFHzD8+v1I2YJ6oXevct1YeS0o9HZyN1Q9qgCgzUFtdOKLv6IedplqoPkcmF0aYet2PkEDo3MlTBckFXPITAMzF8dJSIFo9D8HfdOV0IAdx4O7PtixWKn5y2hMNG0zQPyUecp4pzC6kivAIhyfHilFR61RGL+GPXQ2MWZWFYbAGjyiYJnAmCP3NOTd0jMZEnDkbUvxhMmBYSdETk1rRgm+R4LOzFUGaHqHDLKLX+FIPKcF96hrucXzcWyLbIbEgE98OHlnVYCzRdK8jlqm8tehUc9c9WhQ==" > /home/vagrant/.ssh/authorized\_keys

RUN chmod 600 /home/vagrant/.ssh/authorized\_keys

RUN chown -R vagrant:vagrant /home/vagrant/.ssh

You can log in with the root user but the password wasn't assigned. You can change the password adding a similar line but changing vagrant:vagrant to root:THEPASSWORDYOUCHOOSE or after log in.

**Vagrantfile**

Now create a Vagrantfile:

$ touch Vagrantfile

And add the following content:

Vagrant.configure("2") do |config|

config.vm.provider :docker do |d|

d.build\_dir = "."

d.remains\_running = true

d.has\_ssh = true

end

end

Here you tell Vagrant to build the Docker image from the Dockerfile and the container can be accessed through SSH and must be always running.

d.build\_dir = "."

d.remains\_running = true

d.has\_ssh = true

For installing software you can use a shell script or any provisioning tool supported by Vagrant. You can also add the instructions for installing and configuring the tools you required in the Dockerfile.

If you want to use a bash script, just add the following line after config.vm.provider.

config.vm.provision :shell, path: "script.sh", privileged: false

The privileged option is set to false as you will not require to run these commands with root permissions.

**Up and running**

When running vagrant up, Vagrant will build the Docker image based on the Dockerfile and run the container.

You can log in to the virtual environment running vagrant ssh.

If you want to stop the environment, run vagrant halt. For destroying the virtual environment run vagrant destroy.