

Managing Virtual Machines with Vagrant

let's see how **Vagrant** helps us manage Virtual Machines:

- **Vagrantfile**

It is a text file with the **Ruby** syntax, which has all the information about configuring and provisioning a set of machines. It has details like the machine type, image, networking, provider-specific information, provisioner details, etc. We provide a sample **Vagrantfile** below:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
```

```
Vagrant.configure(2) do |config|
  # Every Vagrant development environment requires a box. You
  can search for
  # boxes at https://atlas.hashicorp.com/search.
  config.vm.box = "centos/7"

  # 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"

  # config.vm.synced_folder "../data", "/vagrant_data"

  config.vm.provider "virtualbox" do |vb|
    # Customize the amount of memory on the VM:
    vb.memory = "1024"
  end

  config.vm.provision "shell", inline: <<-SHELL
    yum install vim -y
  SHELL
end
```

The **vagrant** command reads the configuration given in the configuration file and does different operations, like **up**, **ssh**, **destroy**, etc. The **vagrant** command also has sub-commands like **box** to manage **Box** images, **rdp** to connect to VMs

using **RDP** (**R**emote **D**esktop **P**rotocol), etc. A detailed list of commands is available at its [documentation](#).

- **Boxes**

We need to provide an image in the **Vagrantfile**, which we can use to instantiate machines. In the example above, we have used `centos/7as` the base image. If the image is not available locally, then it can be downloaded from a central repository like [Atlas](#), which is the image repository provided by **Hashicorp**. We can version these images and use them depending on our need, by updating the **Vagrantfile** accordingly.

- **Vagrant Providers**

[Providers](#) are the underlying engine/hypervisor used to provision a machine. By default, **Vagrant** supports **VirtualBox**, **Hyper-V** and **Docker**. We also have custom providers, like **KVM**, **AWS**, etc. **VirtualBox** is the default provider.

- **Synced Folders**

With the *Synced Folder* feature, we can sync a directory on the host system with a VM, which helps the user manage shared files/directories easily. For example, in the above example, if we un-comment the line below from **Vagrantfile**, then the `../data` folder from the current working directory of the host system would be shared with the `/vagrant_data` file on the VM.

```
# config.vm.synced_folder "../data", "vagrant_data"
```

- **Provisioning**

[Provisioners](#) allow us to automatically install software, make configuration changes, etc. after the machine is booted. It is a part of the `vagrant up` process. There are many types of provisioners available, such as **File**, **Shell**, **Ansible**, **Puppet**, **Chef**, **Docker**, etc. In the example below, we used **Shell** as the provisioner to install the `vim` package.

```
config.vm.provision "shell", inline: <<-SHELL
    yum install vim -y
SHELL
```

- **Plugins**

We can use [Plugins](#) to extend the functionality of **Vagrant**.