1. Overview

- Vagrant is a tool for building and managing virtual machine environments in a single workflow.
- Lowers development environment setup time, increases production parity, and makes the “works on my machine” excuse a relic of the past.

“Flexibility joined with consistency”

2. Requirements

- Vagrant supports both Windows and Linux operating systems.
- It does not require any pre-requisite for installation. However, in most cases, it is used alongside Oracle VirtualBox for managing the virtual machines.

3. Architecture

- Uses Provisioners and Providers as building blocks to manage the development environments.
  - Provisioners allow users to customize the configuration of virtual environments. Puppet and Chef are the two most widely used provisioners in the Vagrant ecosystem.
  - Providers are the services that Vagrant uses to set up and create virtual environments.
- Sits on top of virtualization software as a wrapper and helps the developer interact easily with the providers.
- Automates the configuration of virtual environments using Chef or Puppet, and the user does not have to directly use any other virtualization software.
- Machine and software requirements are written in a file called “Vagrantfile” to execute necessary steps in order to create a development-ready box.
- Box is a format and an extension (.box) for Vagrant environments that is copied to another machine in order to replicate the same environment.

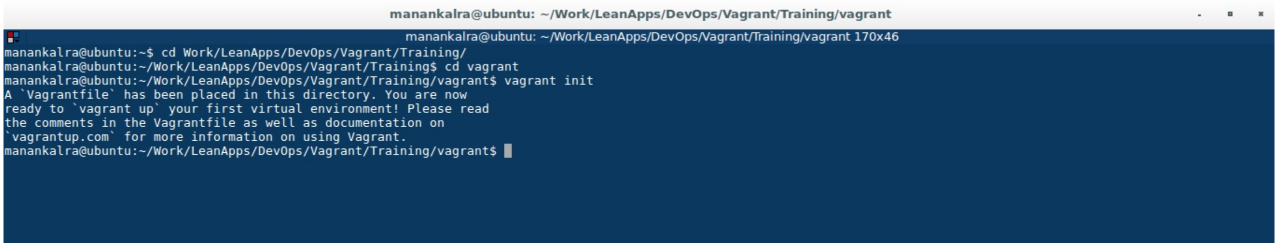
4. Installation for Linux (Ubuntu 16.04 LTS)

- Open Terminal. Run `sudo apt-get update` to update repository information.
- For installing Vagrant, execute `sudo apt-get install vagrant`.
- Additionally, run the `vagrant --version` command to check the version and verify proper installation of Vagrant on your system.
- If VirtualBox isn't already installed, run `sudo apt-get install virtualbox` to install it.

5. Hands-on

a. Initialization

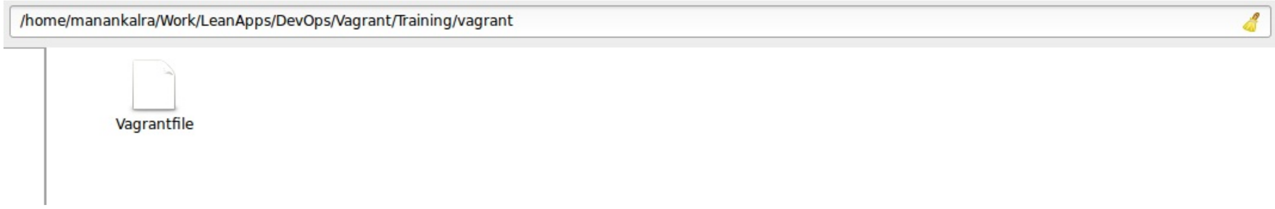
- The first step to begin working with Vagrant is initializing it to generate a *Vagrantfile*. This is done by running the command `vagrant init` in the destination folder. This will generate a text file named 'Vagrantfile' in the current directory.



manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant
manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant 170x46

```
manankalra@ubuntu:~$ cd Work/LeanApps/DevOps/Vagrant/Training/
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training$ cd vagrant
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ 
```

The *Vagrantfile* is a configuration file containing information about the settings and configurations of the VM. It needs to be generated every time a new VM is to be created by Vagrant.



/home/manankalra/Work/LeanApps/DevOps/Vagrant/Training/vagrant
Vagrantfile

- Make changes to the *Vagrantfile* according to the configuration needs for the VM.

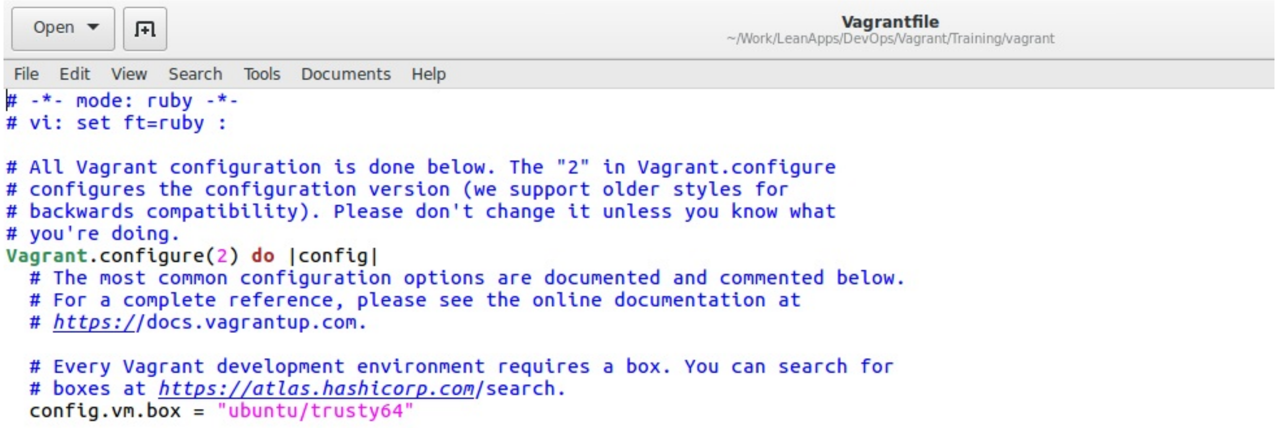
b. Configuration

i. Vagrantfile

- Describes the type of machine required for a project, and how to configure and provision these machines.
- Only one Vagrantfile is needed per project.

ii. Boxes

- Boxes are the package format for Vagrant environments.
- A box can be used by anyone on any platform that Vagrant supports to bring up an identical working environment.
- Edit the *box* according to your choice in the *Vagrantfile*. A search index for boxes can be found here.



```
# -*- 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 = "ubuntu/trusty64"
```

iii. Provisioning

- Provisioners in Vagrant allow you to automatically install software, alter configurations, and more.
- Following is an example to provision the installation of Apache Server and git on the machine as a part of the `vagrant up` process:
    - Edit the *Vagrantfile* to add this snippet which is an inline shell command to trigger the required installation.

    ```
    config.vm.provision "shell", inline: <<-SHELL
        sudo apt-get update
        sudo apt-get install -y apache2
    SHELL
    ```

    - This snippet can be added to use an external shell script and provision git installation.

      ```
      config.vm.provision "shell", path: "provision-git.sh"
      ```

      Here, *provision-git.sh* contains a shell script to install git.

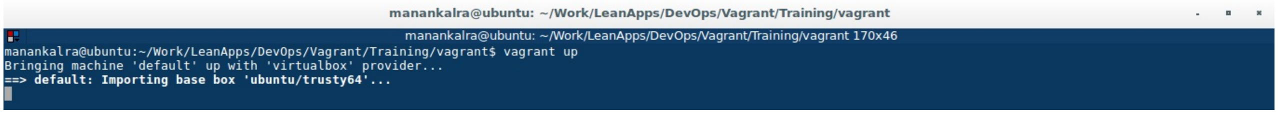
      ```
      #!/usr/bin/env bash
      echo "Installing git..."
      sudo apt-get install -y git
      ```

- Now `vagrant up` will install all the configured packages automatically.

Other configuration settings may involve: Networking, Synced Folders, Multi-Machine, Providers, Plugins, Push, Share etc.

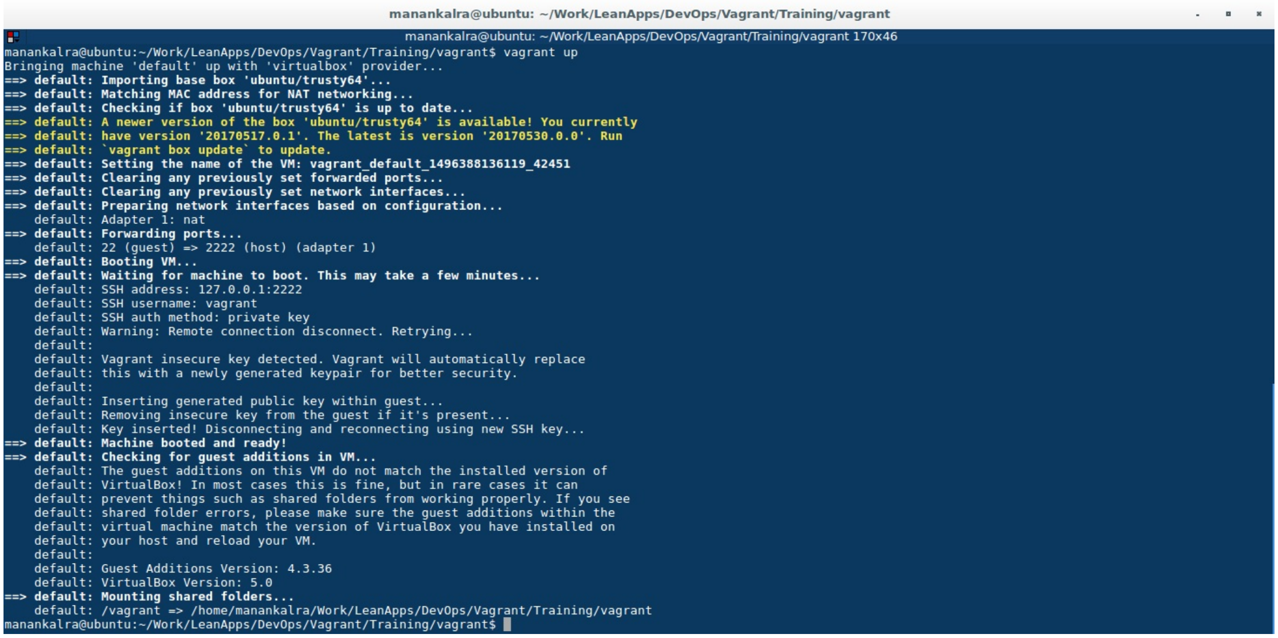
c. Running the VM

- After the *Vagrantfile* has been changed to contain the required configurations for the new VM, execute **`vagrant up`** from the same directory where the 'Vagrantfile' was generated.


manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant
manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant 170x46

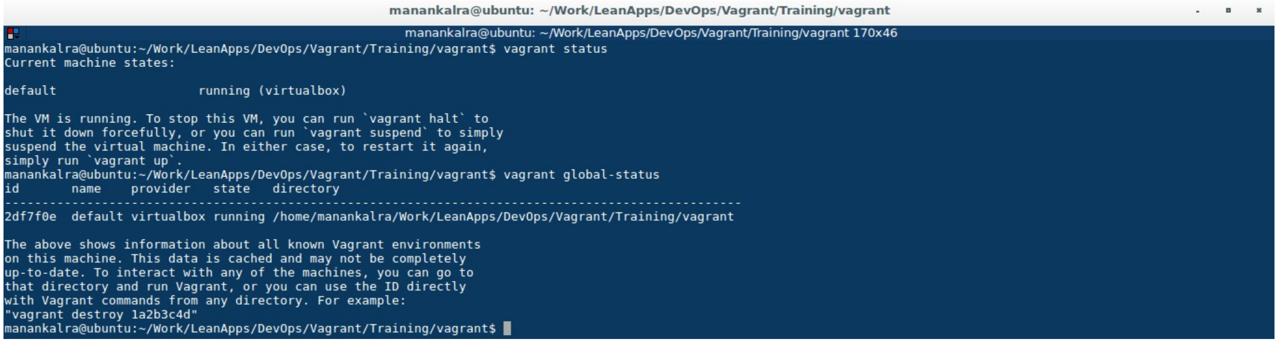
```
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
```

This will start the process of fetching the box file for VM creation, creating the VM with the stated configurations, and booting it automatically after creation.


manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant
manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant 170x46

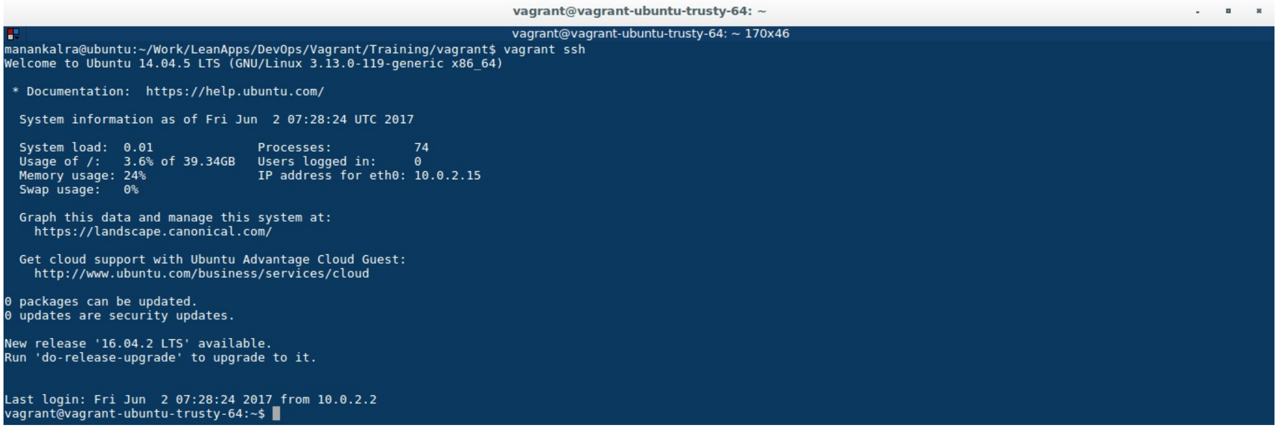
```
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' is up to date...
==> default: A newer version of the box 'ubuntu/trusty64' is available! You currently
==> default: have version '20170517.0.1'. The latest is version '20170530.0.0'. Run
==> default: `vagrant box update` to update.
==> default: Setting the name of the VM: vagrant_default_1496388136119_42451
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default: Warning: Remote connection disconnect. Retrying...
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you see
    default: shared folder errors, please make sure the guest additions within the
    default: virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default:
    default: Guest Additions Version: 4.3.36
    default: VirtualBox Version: 5.0
==> default: Mounting shared folders...
    default: /vagrant => /home/manankalra/Work/LeanApps/DevOps/Vagrant/Training/vagrant
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$
```

- Status of the created VM can be checked by executing `vagrant status` or `vagrant global-status`


manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant
manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant 170x46

```
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ vagrant status
Current machine states:

default                   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ vagrant global-status
id       name    provider   state   directory
--------------------------------------------------------------------------------------------------
2df7f0e  default virtualbox running /home/manankalra/Work/LeanApps/DevOps/Vagrant/Training/vagrant

The above shows information about all known Vagrant environments
on this machine. This data is cached and may not be completely
up-to-date. To interact with any of the machines, you can go to
that directory and run Vagrant, or you can use the ID directly
with Vagrant commands from any directory. For example:
"vagrant destroy 1a2b3c4d"
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ █
```

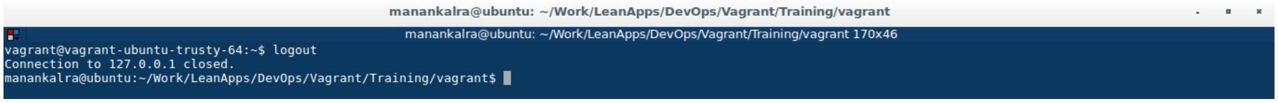
- To enter the newly created VM, execute `vagrant ssh`.


vagrant@vagrant-ubuntu-trusty-64: ~
vagrant@vagrant-ubuntu-trusty-64: ~ 170x46

```
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ vagrant ssh
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-119-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Fri Jun  2 07:28:24 UTC 2017

  System load:  0.01              Processes:           74
  Usage of /:   3.6% of 39.34GB   Users logged in:     0
  Memory usage: 24%               IP address for eth0: 10.0.2.15
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '16.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Jun  2 07:28:24 2017 from 10.0.2.2
vagrant@vagrant-ubuntu-trusty-64:~$ █
```

Commands can be executed in the new VM from this shell prompt. Alternatively, you can also enter the new VM via Virtualbox.

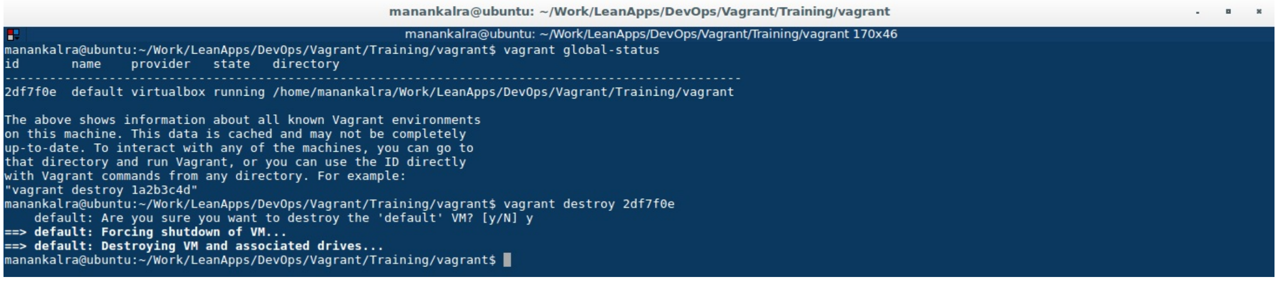
- At any point, run `logout` command in the shell prompt of the VM to exit from the VM and return to the host machine where you installed Vagrant.


manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant
manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant 170x46

```
vagrant@vagrant-ubuntu-trusty-64:~$ logout
Connection to 127.0.0.1 closed.
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ █
```

d. Destroying the VM

- Execute `vagrant destroy <id>`.


manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant
manankalra@ubuntu: ~/Work/LeanApps/DevOps/Vagrant/Training/vagrant 170x46

```
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ vagrant global-status
id       name    provider   state   directory
--------------------------------------------------------------------------------------------------
2df7f0e  default virtualbox running /home/manankalra/Work/LeanApps/DevOps/Vagrant/Training/vagrant

The above shows information about all known Vagrant environments
on this machine. This data is cached and may not be completely
up-to-date. To interact with any of the machines, you can go to
that directory and run Vagrant, or you can use the ID directly
with Vagrant commands from any directory. For example:
"vagrant destroy 1a2b3c4d"
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ vagrant destroy 2df7f0e
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Forcing shutdown of VM...
==> default: Destroying VM and associated drives...
manankalra@ubuntu:~/Work/LeanApps/DevOps/Vagrant/Training/vagrant$ █
```

6. List of commands

- `--version` : print the version and exit
- `box` : manages boxes: installation, removal etc..
- `destroy` : stops and deletes all traces of the vagrant machine
- `global-status` : outputs status Vagrant envisonments for the user
- `halt` : stops the vagrant machine
- `help` : shows the help for a subcommand
- `init` : initializes a new Vagrant environment by generating a Vagrantfile
- `login` : login to Hashicorp’s Atlas
- `package` : packages a running vagrant environment into a box
- `plugin` : manages plugins: install, uninstall, update etc.
- `port` : displays information about guest port mappings
- `powershell` : connects to mahcine via powershell remoting
- `provison` : provisions the vagrant machine
- `push` : deploys code in this environment to a configured destination
- `sdp` : connects to machine via RDP
- `reload`: restarts vagrant machine, loads new Vagrantfile configuration
- `resume` : resume a suspended vagrant machine
- `snapshot` : manages snapshot: saving, restoring etc.
- `ssh` : connects to machine via SSH
- `ssh-config` : outputs OpenSSH valid configuration to connect to the machine
- `status` : outputs status of the vagrant machine
- `suspend` : suspends the machine
- `up` : starts and provisions the vagrant environment

Run `vagrant list-commands` for additional and more advanced subcommands.