**What is Vagrant?** Vagrant is a simple and easy to use tool that makes it really easy to manage virtual machines from the command-line interface.

Vagrant out of the box supports VirtualBox, Hyper-V, Docker and has the ability to manage other types of machines, like VMWare or Amazon EC2, by using other [providers](https://www.vagrantup.com/docs/providers).

In this tutorial i will show how to initialize a new Vagrant environment, download, provision and start a virtual machine, how to SSH into it and how to stop or destroy it after.

This tutorial supposes that you have already installed one of supported virtual providers, like VirtualBox or Docker.

Install Vagrant

Download and install the latest version of Vagrant package from the official [download page](https://www.vagrantup.com/downloads.html).

Print the Vagrant’s version to make sure that you have successfully installed it:

$ vagrant -v

Vagrant 2.0.0

Vagrant Boxes

Standard templates of the virtual machines in Vagrant are called boxes.

You can find a public list of Vagrant boxes on the [box search page](https://app.vagrantup.com/boxes/search).

**VirtualBox ≠ VMWare:** Vagrant boxes are all provider-specific. Not all boxes are available for all providers. You may need to sort by a provider that you have on your local system to narrow down your search.

Once you find a box you want to use – create a directory where you will store your new project:

$ mkdir -p vagrant-projects/tutorial

$ cd vagrant-projects/tutorial

nitializes a new Vagrant environment inside the project’s directory by typing vagrant init <boxpath>, where <boxpath> is the name of Vagrant box.

For example, to use the base image of Ubuntu-16.04, type:

$ vagrant init ubuntu/xenial64

The vagrant init command creates a Vagrantfile in the current directory, that describes the type of machine required for a project and how to configure and provision this machine.

## Vagrant Up

Start Vagrant environment:

$ vagrant up

The vagrant up command creates, configures and starts a virtual machine according to your Vagrantfile.

On the first run it automatically downloads the required Vagrant box from the [box](https://app.vagrantup.com/boxes/search)

[repository](https://app.vagrantup.com/boxes/search) and performs provisioning.

## Vagrant SSH

SSH into a running Vagrant machine and access the shell:

$ vagrant ssh

## Clean Up Vagrant

If you need to suspend the development process – you can either stop the Vagrant machine or just suspend it.

Stop Vagrant machine:

$ vagrant halt

Suspend a machine (remember the state):

$ vagrant suspend

To clean up the development environment – you can destroy the machine.

Stop and delete all traces of the Vagrant machine:

$ vagrant destroy

When you need again a clear Vagrant environment based on a standard template – just run vagrant up.

If you are an **operations engineer** or **DevOps engineer**, Vagrant gives you a disposable environment and consistent workflow for developing and testing infrastructure management scripts. You can quickly test things like shell scripts, Chef cookbooks, Puppet modules, and more using local virtualization such as VirtualBox or VMware. Then, with the same configuration, you can test these scripts on remote clouds such as AWS or RackSpace with the same workflow. Ditch your custom scripts to recycle EC2 instances, stop juggling SSH prompts to various machines, and start using Vagrant to bring sanity to your life.