[Vagrant Documentation](#)

- [Vagrant Book](#)
- [Home](#)

Contents

- [Overview](#)
- [Why Vagrant?](#)
- [Installation](#)
- [Getting Started](#)
- [Command-Line Interface](#)
- [Vagrant Share](#)
- [Vagrantfile](#)
- [Boxes](#)
- [Provisioning](#)
- [Networking](#)
- [Synced Folders](#)
- [Multi-Machine](#)
- [Providers](#)
- [Plugins](#)
- [Push](#)
- [Other](#)

Why Vagrant?

Vagrant provides easy to configure, reproducible, and portable work environments built on top of industry-standard technology and controlled by a single consistent workflow to help maximize the productivity and flexibility of you and your team.

To achieve its magic, Vagrant stands on the shoulders of giants. Machines are provisioned on top of VirtualBox, VMware, AWS, or [any other provider](#). Then, industry-standard [provisioning tools](#) such as shell scripts, Chef, or Puppet, can be used to automatically install and configure software on the machine.

How Vagrant Benefits You

If you're a **developer**, Vagrant will isolate dependencies and their configuration within a single disposable, consistent environment, without sacrificing any of the tools you're used to working with (editors, browsers, debuggers, etc.). Once you or someone else creates a single [Vagrantfile](#), you just need to `vagrant up` and everything is installed and configured for you to work. Other members of your team create their development environments from the same configuration, so whether you're working on Linux, Mac OS X, or Windows, all your team members are running code in the same environment, against the same dependencies, all configured the same way. Say goodbye to "works on my machine" bugs.

If you're an **operations 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.

If you're a **designer**, Vagrant will automatically set everything up that is required for that web app in order for you to focus on doing what you do best: design. Once a developer configures Vagrant, you don't need to worry about how to get that app running ever again. No more bothering other developers to help you fix your environment so you can test designs. Just check out the code, `vagrant up`, and start designing.

- [Documentation](#)
- [About](#)
- [Support](#)
- [Download](#)

-
- by
-

Copyright © 2015 HashiCorp