Getting started with Vagrant in PhpStorm

Tweet

This tutorial describes how to get started with Vagrant integration in PhpStorm.

1. Ensure Vagrant and Oracle VirtualBox are on your system

Vagrant is a tool which helps us create reproducible development environments. By scripting a virtual machine, Vagrant allows us to share a development machine by sharing a simple script. Other developers can then launch an identical machine by running the vagrant up command. PhpStorm comes with support for Vagrant within the IDE.

To be able to work with Vagrant in PhpStorm, both Vagrant and Oracle VirtualBox have to be installed on your system.
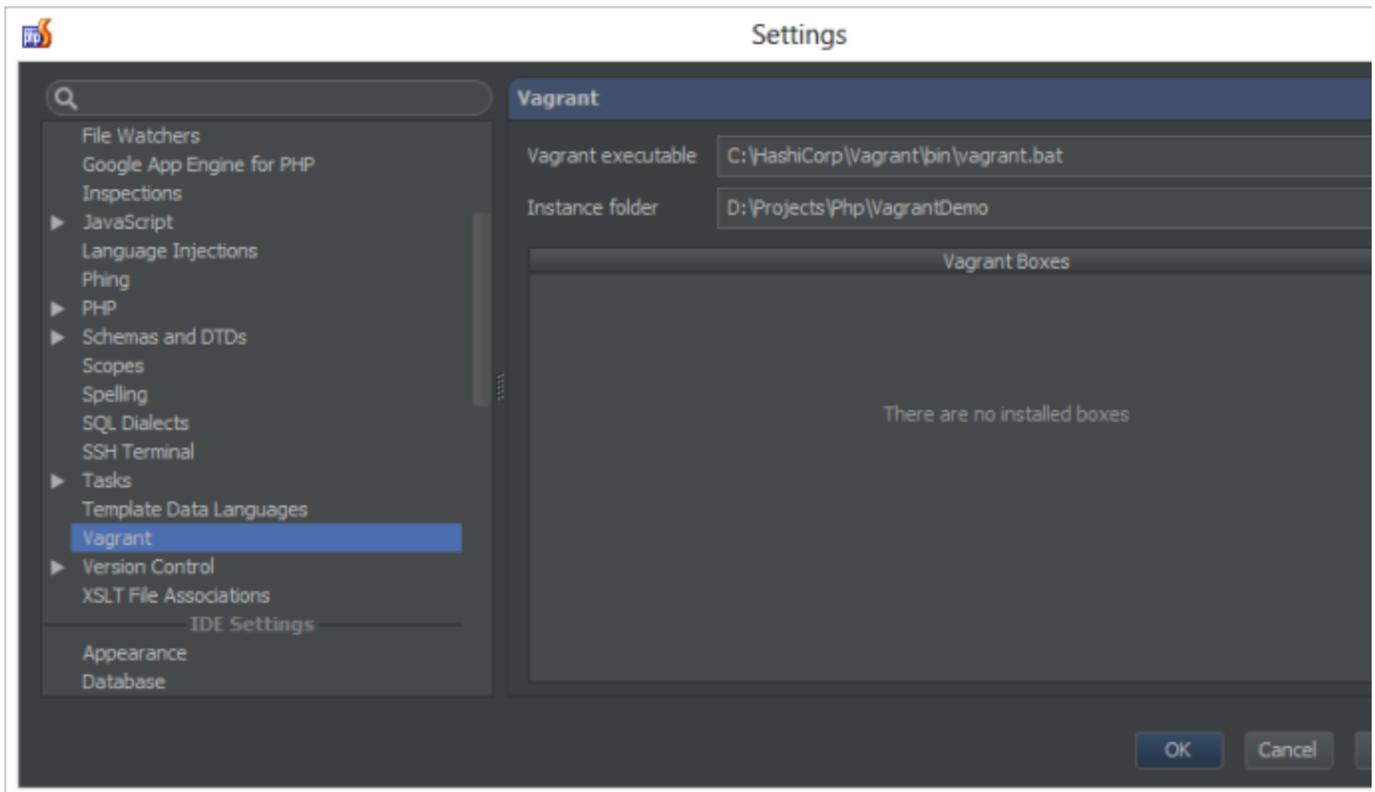
1.1. Download and install Vagrant

Download and install Vagrant from their official website.
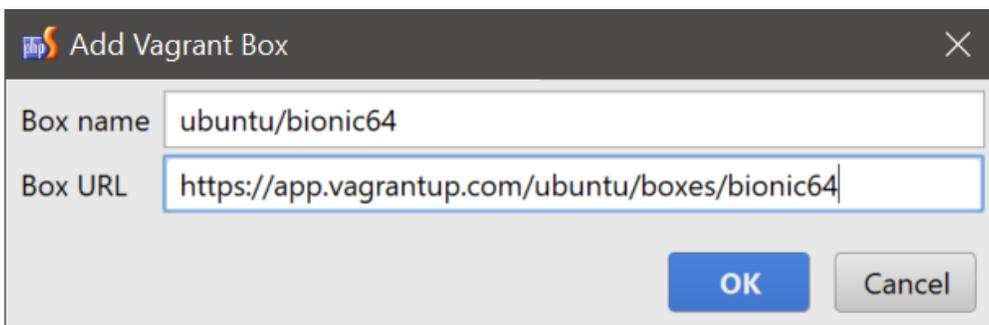
1.2. Download and install Oracle VirtualBox

From the VirtualBox website, download and install the latest version of VirtualBox.

2. Configure Vagrant project settings

From the Settings | Project Settings | Vagrant window, specify the required parameters for Vagrant integration. The path to the Vagrant executable must be provided, as well as the instance folder (can be empty to use the project directory).

Using the Add button, we can add our first Vagrant Box. PhpStorm will provide the name and URL to the lucid32 box by default; other boxes can be specified as well.
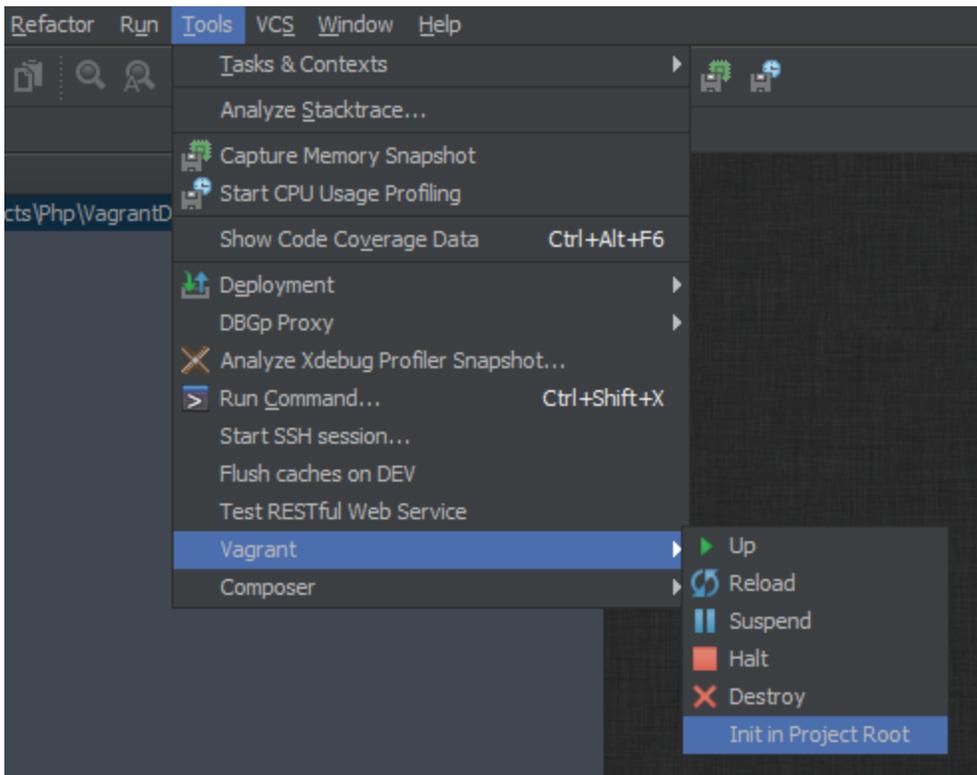

3. Initialize VagrantFile

The VagrantFile contains all information about the virtual machine. It contains details about the virtual machine such as the virtual IP address, port mappings and memory to assign. Next to that, it can specify which folders are shared and which third-party software should be installed on the machine.
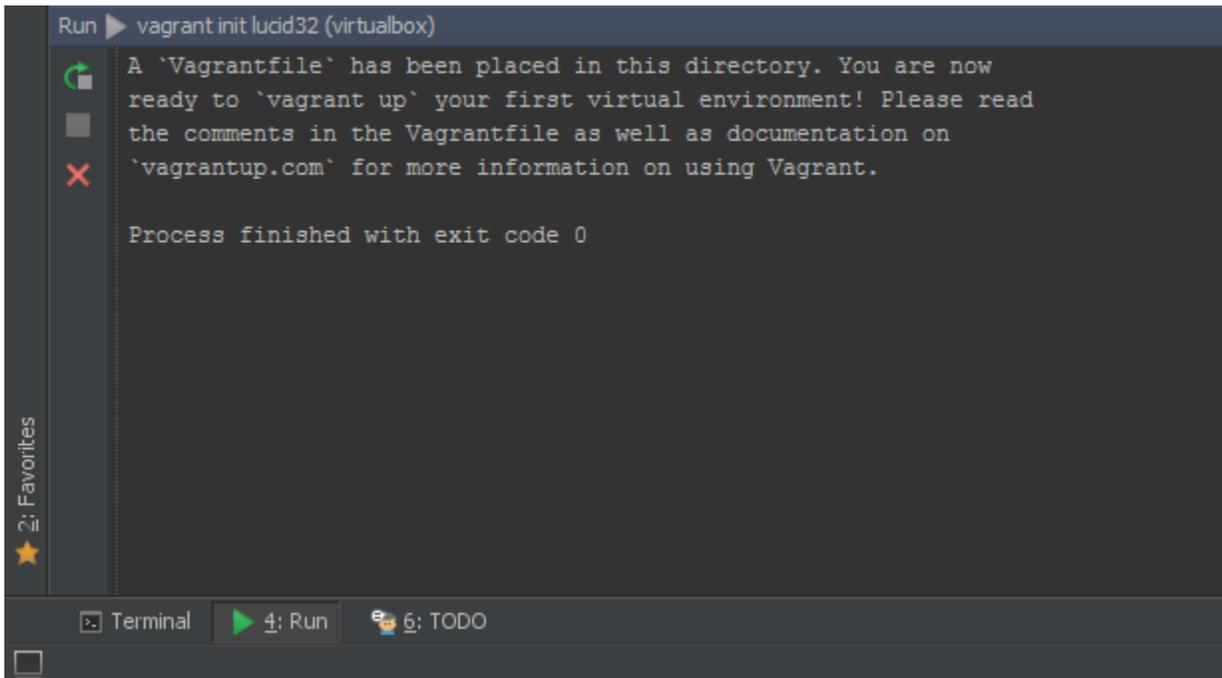
> ⚠️ Note: the PuPHPet website has a great tool for configuring a Vagrant machine and generating a VagrantFile.

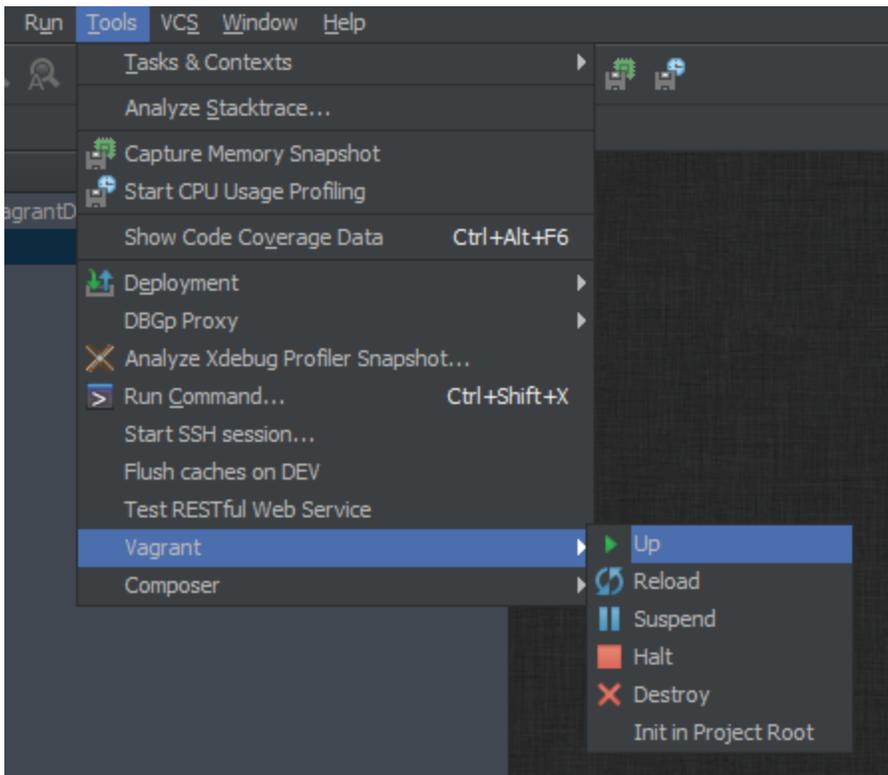The VagrantFile can be created manually or by using the Tools | Vagrant | Init in Project Root menu.

This will trigger creation of a default VagrantFile.

4. Vagrant up

The Vagrant machine can be started from the Tools | Vagrant | Up menu. This will configure the virtual machine in VirtualBox if this hasn't been done yet, as well as boot it.

Progress can be seen from the Run tool window.

```
Run  ▶  vagrant up

    Bringing machine 'default' up with 'virtualbox' provider...
    [default] Box 'lucid32 (virtualbox)' was not found. Fetching box from specified URL for
    the provider 'virtualbox'. Note that if the URL does not have
    a box for this provider, you should interrupt Vagrant now and add
    the box yourself. Otherwise Vagrant will attempt to download the
    full box prior to discovering this error.
    Downloading or copying the box...
     [0KExtracting box...
    Successfully added box 'lucid32 (virtualbox)' with provider 'virtualbox'!
    [default] Importing base box 'lucid32 (virtualbox)'...
     [0K[default] Matching MAC address for NAT networking...
    [default] Setting the name of the VM...
    [default] Clearing any previously set forwarded ports...
    [default] Creating shared folders metadata...
    [default] Clearing any previously set network interfaces...
    [default] Preparing network interfaces based on configuration...
    [default] Forwarding ports...
    [default] -- 22 => 2222 (adapter 1)
    [default] Booting VM...
    [default] Waiting for VM to boot. This can take a few minutes.
    [default] VM booted and ready for use!
    [default] Mounting shared folders...
    [default] -- /vagrant

    Process finished with exit code 0


  ▶ 4: Run      6: TODO      Terminal
```

Once completed, the Vagrant machine will be ready for use.

> ⊘ You can now continue to Working with Advanced Vagrant features in PhpStorm.

Tweet