Laravel Framework for Web Applications

SESSION 01

# SETTING UP THE ENVIRONMENT

# OBJECTIVES

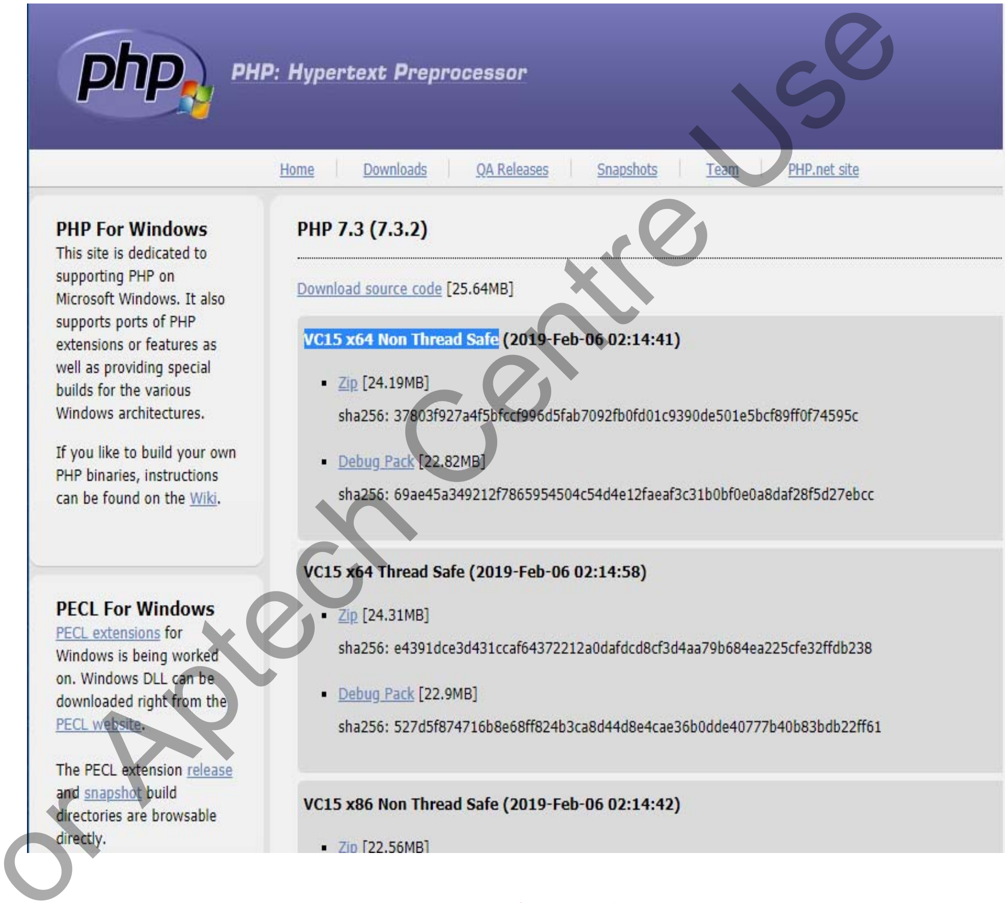
- Identify components to set up the development environment
- Learn to set up PHP and Composer
- Understand the basics of VirtualBox, Vagrant, Homestead, and Laravel
- Learn how to verify installations and versions of various software

# SETTING UP THE ENVIRONMENT

## List of components

- PHP
- Composer
- VirtualBox
- Vagrant
- Git Bash
- Homestead

# SETTING UP PHP



PHP: Hypertext Preprocessor
Home | Downloads | QA Releases | Snapshots | Team | PHP.net site

**PHP For Windows**

This site is dedicated to supporting PHP on Microsoft Windows. It also supports ports of PHP extensions or features as well as providing special builds for the various Windows architectures.

If you like to build your own PHP binaries, instructions can be found on the Wiki.

**PECL For Windows**

PECL extensions for Windows is being worked on. Windows DLL can be downloaded right from the PECL website.

The PECL extension release and snapshot build directories are browsable directly.

**PHP 7.3 (7.3.2)**

Download source code [25.64MB]

**VC15 x64 Non Thread Safe (2019-Feb-06 02:14:41)**

- Zip [24.19MB]

  sha256: 37803f927a4f5bfccf996d5fab7092fb0fd01c9390de501e5bcf89ff0f74595c

- Debug Pack [22.82MB]

  sha256: 69ae45a349212f7865954504c54d4e12faeaf3c31b0bf0e0a8daf28f5d27ebcc

**VC15 x64 Thread Safe (2019-Feb-06 02:14:58)**

- Zip [24.31MB]

  sha256: e4391dce3d431ccaf64372212a0dafdcd8cf3d4aa79b684ea225cfe32ffdb238

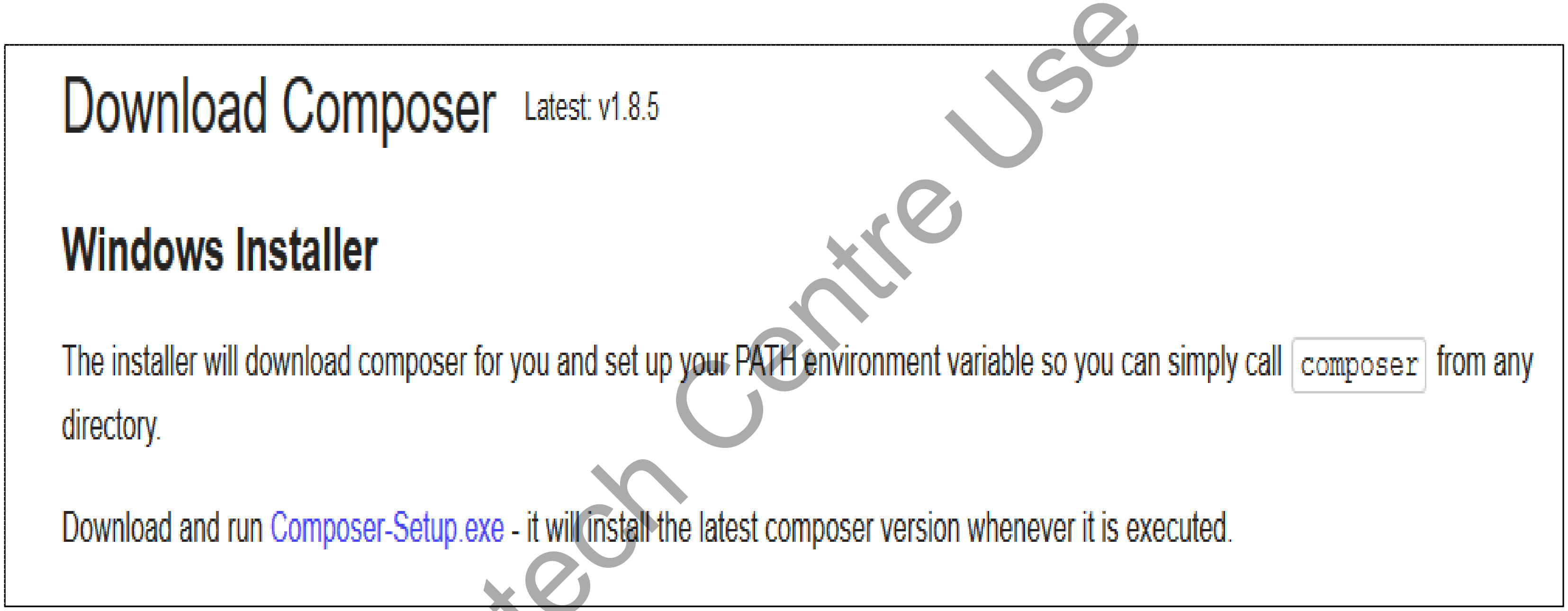
- Debug Pack [22.9MB]

  sha256: 527d5f874716b8e68ff824b3ca8d44d8e4cae36b0dde40777b40b83bdb22ff61

**VC15 x86 Non Thread Safe (2019-Feb-06 02:14:42)**

- Zip [22.56MB]

# SETTING UP COMPOSER (1-3)



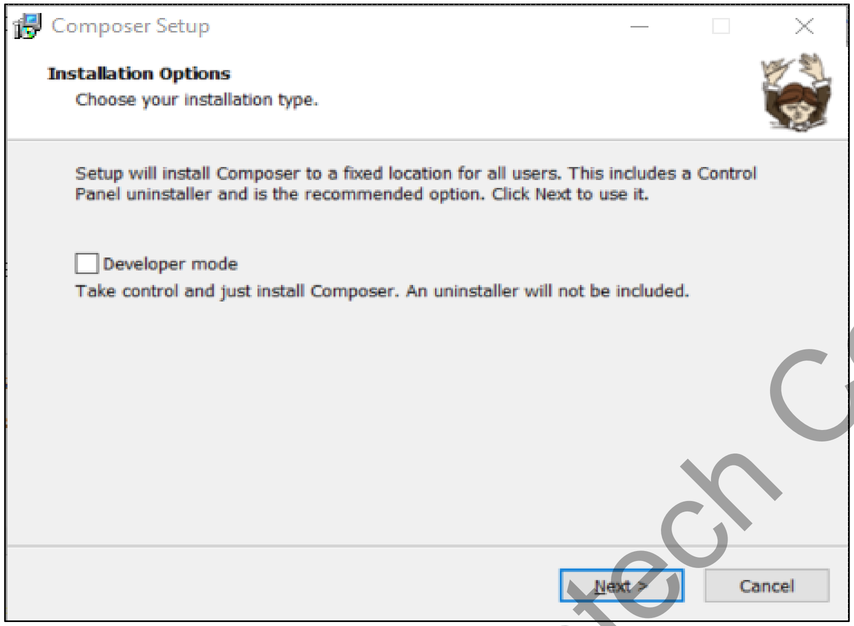
Download Composer Latest: v1.8.5

Windows Installer

The installer will download composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run Composer-Setup.exe - it will install the latest composer version whenever it is executed.

# SETTING UP COMPOSER (2-3)



Composer Setup

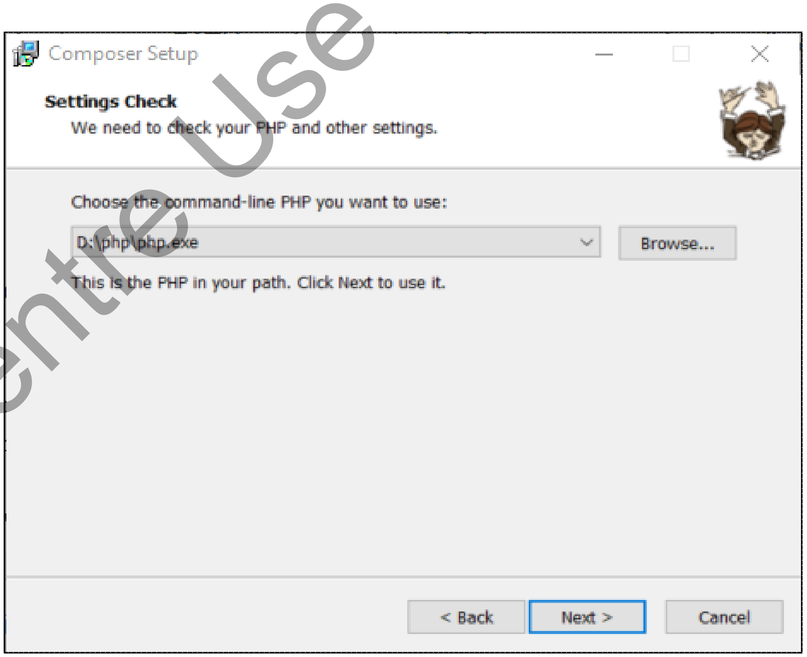
**Installation Options**

Choose your installation type.

Setup will install Composer to a fixed location for all users. This includes a Control Panel uninstaller and is the recommended option. Click Next to use it.

□ Developer mode

Take control and just install Composer. An uninstaller will not be included.



Composer Setup

**Settings Check**

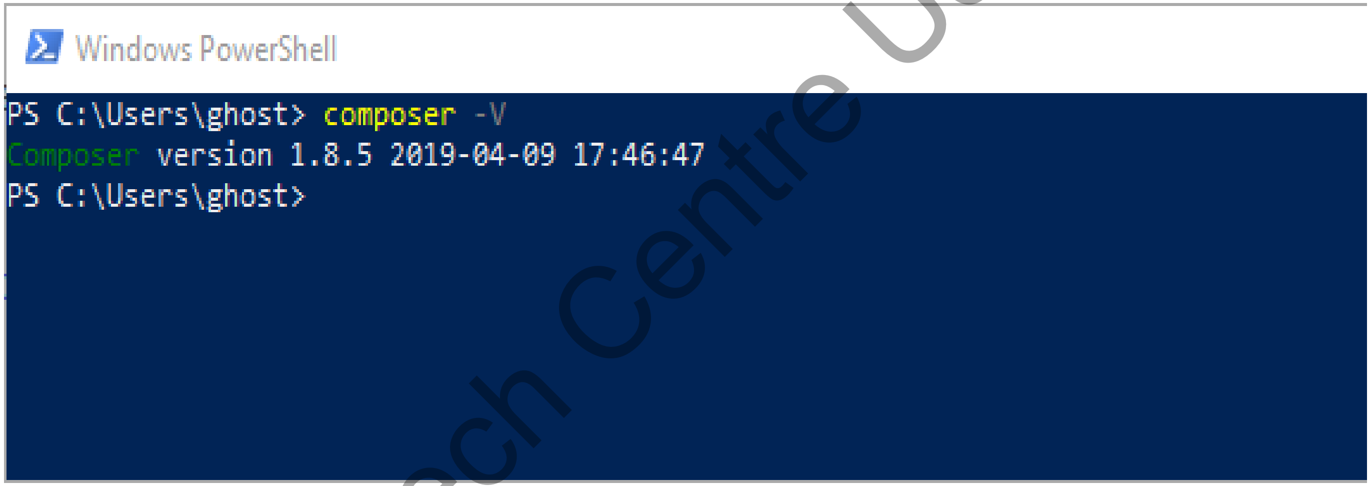
We need to check your PHP and other settings.

Choose the command-line PHP you want to use:

D:\php\php.exe

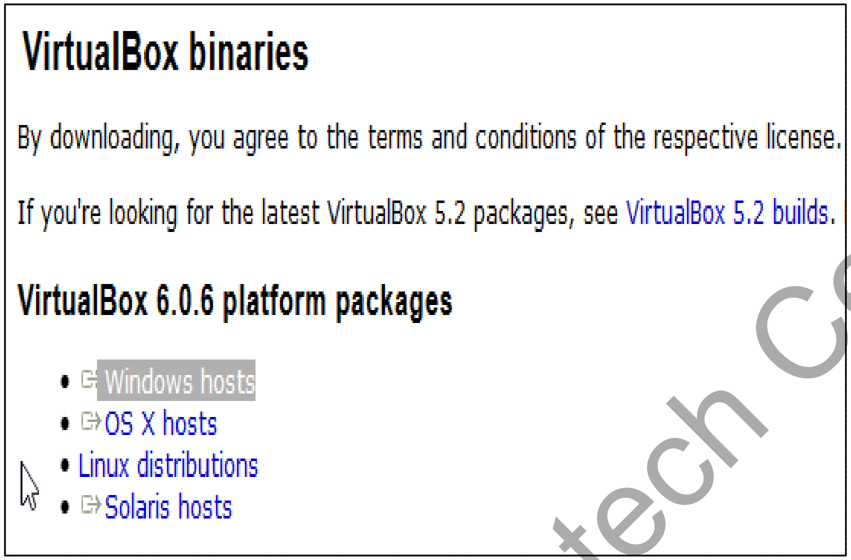
This is the PHP in your path. Click Next to use it.

# SETTING UP COMPOSER (3-3)



```
PS C:\Users\ghost> composer -V
Composer version 1.8.5 2019-04-09 17:46:47
PS C:\Users\ghost>
```

# INSTALLING VIRTUALBOX (1-3)



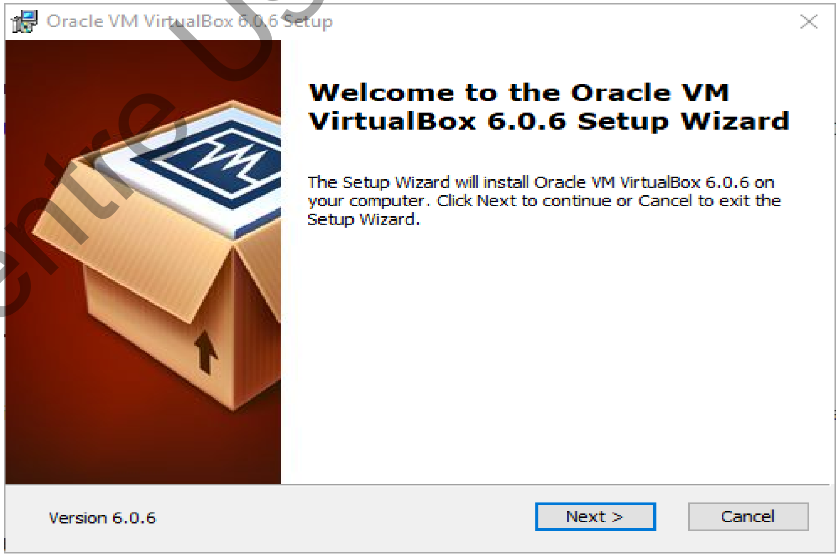
## VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 5.2 packages, see VirtualBox 5.2 builds.

### VirtualBox 6.0.6 platform packages

- Windows hosts
- OS X hosts
- Linux distributions
- Solaris hosts

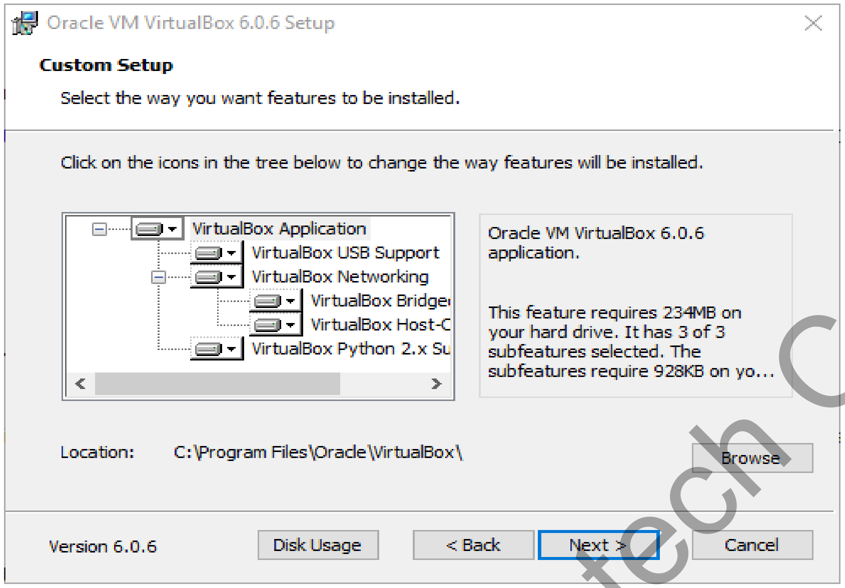


**Welcome to the Oracle VM VirtualBox 6.0.6 Setup Wizard**

The Setup Wizard will install Oracle VM VirtualBox 6.0.6 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Version 6.0.6

# INSTALLING VIRTUALBOX (2-3)


Oracle VM VirtualBox 6.0.6 Setup

**Custom Setup**

Select the way you want features to be installed.

Click on the icons in the tree below to change the way features will be installed.

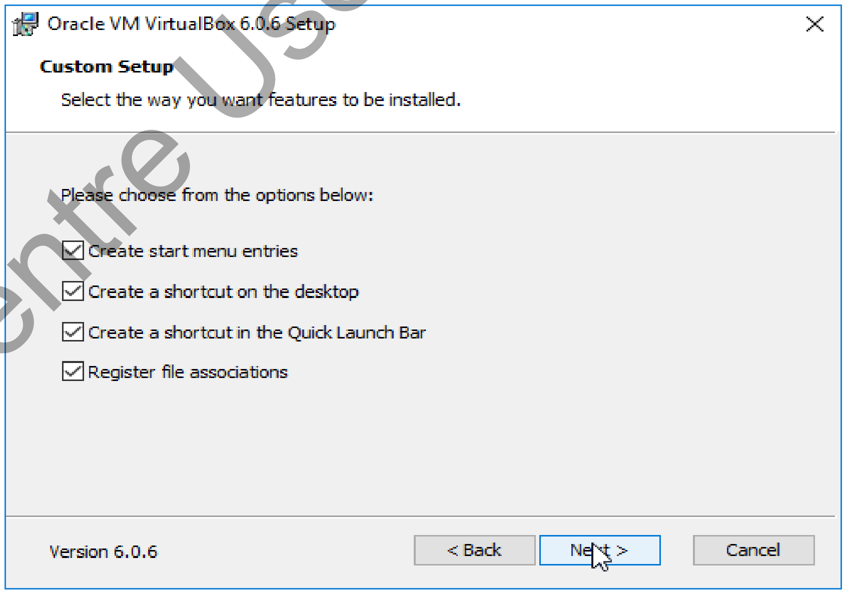
- VirtualBox Application
  - VirtualBox USB Support
  - VirtualBox Networking
    - VirtualBox Bridge
    - VirtualBox Host-C
  - VirtualBox Python 2.x Su

Oracle VM VirtualBox 6.0.6 application.

This feature requires 234MB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 928KB on yo...

Location: C:\Program Files\Oracle\VirtualBox\

Version 6.0.6


Oracle VM VirtualBox 6.0.6 Setup

**Custom Setup**

Select the way you want features to be installed.

Please choose from the options below:

☑ Create start menu entries

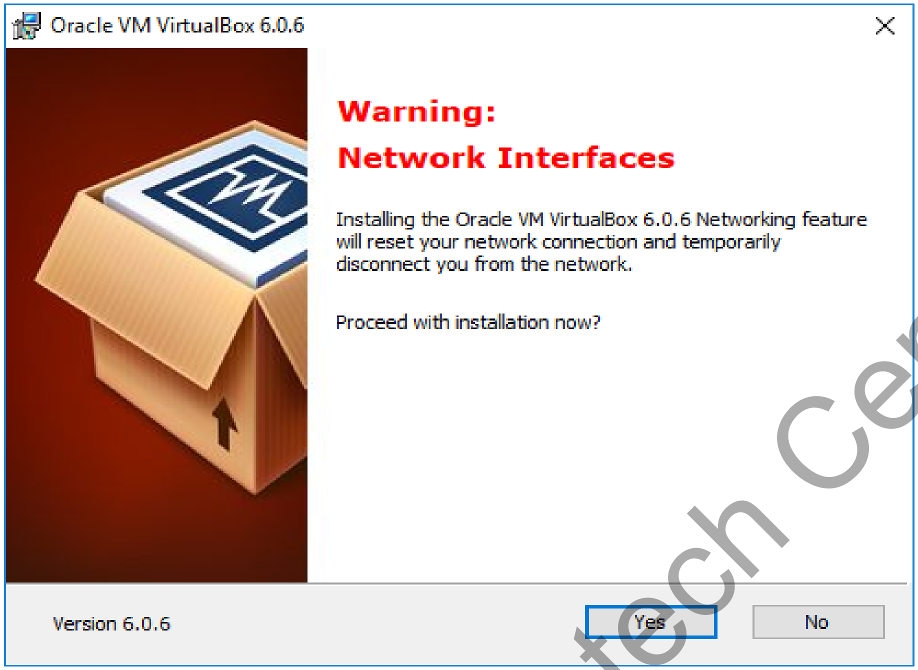
☑ Create a shortcut on the desktop

☑ Create a shortcut in the Quick Launch Bar

☑ Register file associations

Version 6.0.6

# INSTALLING VIRTUALBOX (3-3)



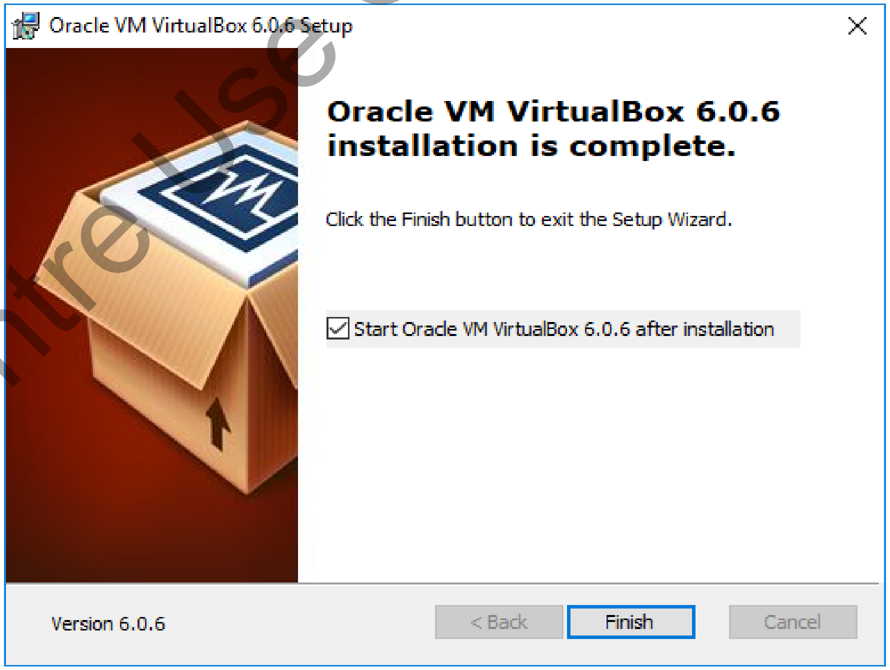
Oracle VM VirtualBox 6.0.6

**Warning:**
**Network Interfaces**

Installing the Oracle VM VirtualBox 6.0.6 Networking feature will reset your network connection and temporarily disconnect you from the network.

Proceed with installation now?

Version 6.0.6



Oracle VM VirtualBox 6.0.6 Setup

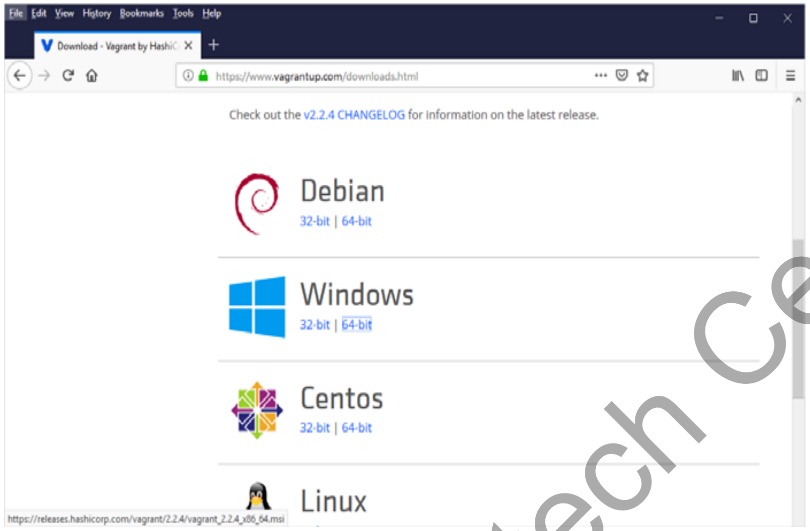
**Oracle VM VirtualBox 6.0.6 installation is complete.**

Click the Finish button to exit the Setup Wizard.

☑ Start Oracle VM VirtualBox 6.0.6 after installation

Version 6.0.6

# INSTALLING VAGRANT (1-4)


File Edit View History Bookmarks Tools Help
Download - Vagrant by HashiC
https://www.vagrantup.com/downloads.html

Check out the v2.2.4 CHANGELOG for information on the latest release.

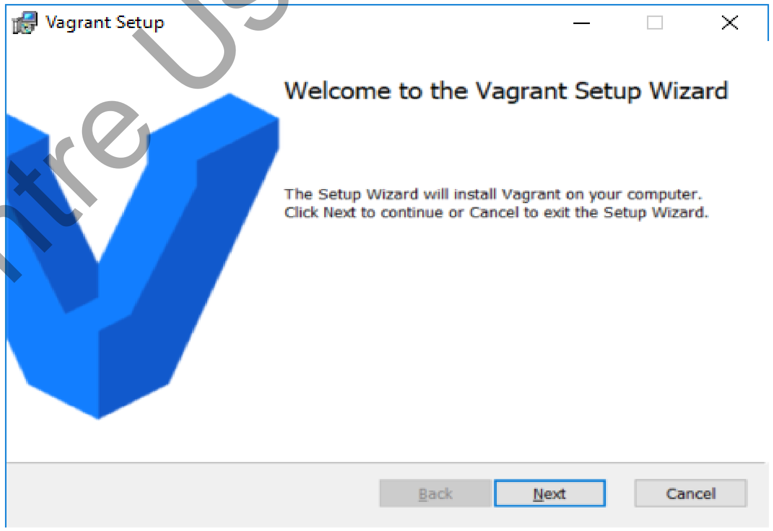
Debian
32-bit | 64-bit

Windows
32-bit | 64-bit

Centos
32-bit | 64-bit

Linux

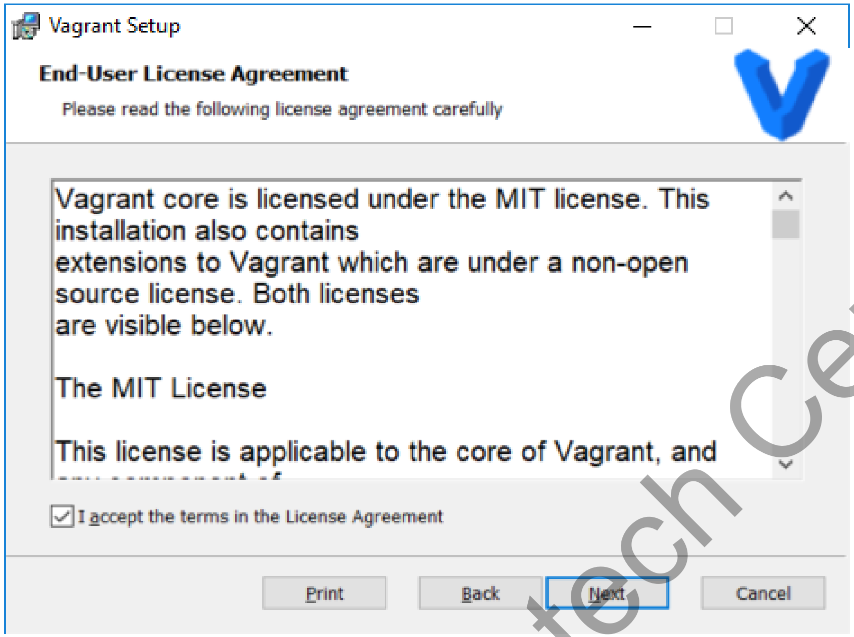
https://releases.hashicorp.com/vagrant/2.2.4/vagrant_2.2.4_x86_64.msi



Welcome to the Vagrant Setup Wizard

The Setup Wizard will install Vagrant on your computer.
Click Next to continue or Cancel to exit the Setup Wizard.

# INSTALLING VAGRANT (2-4)



**End-User License Agreement**

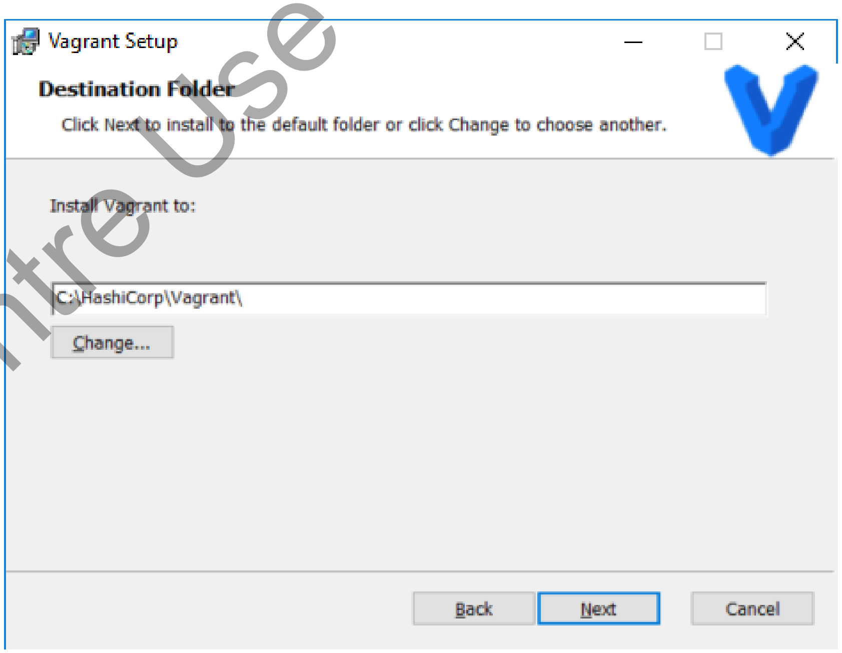
Please read the following license agreement carefully

Vagrant core is licensed under the MIT license. This installation also contains extensions to Vagrant which are under a non-open source license. Both licenses are visible below.

The MIT License

This license is applicable to the core of Vagrant, and

☑ I accept the terms in the License Agreement



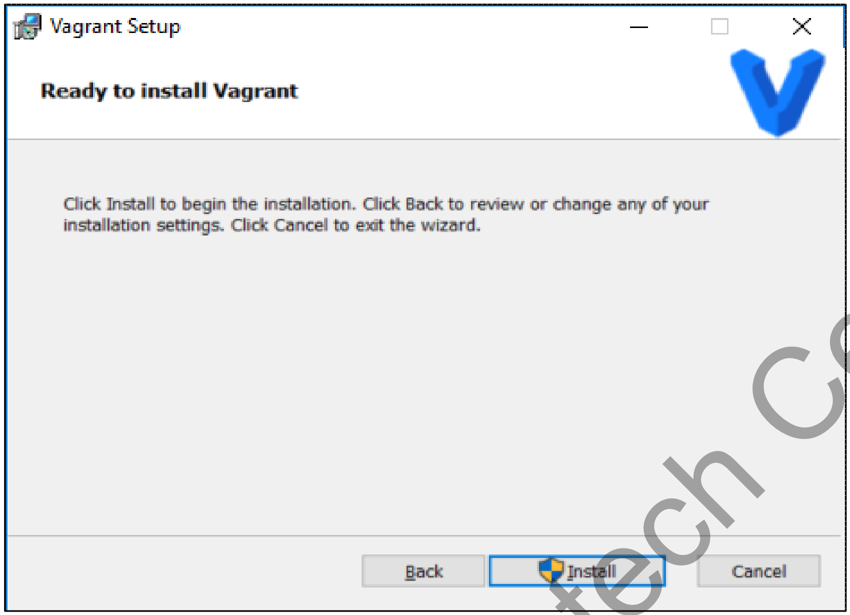
**Destination Folder**

Click Next to install to the default folder or click Change to choose another.

Install Vagrant to:

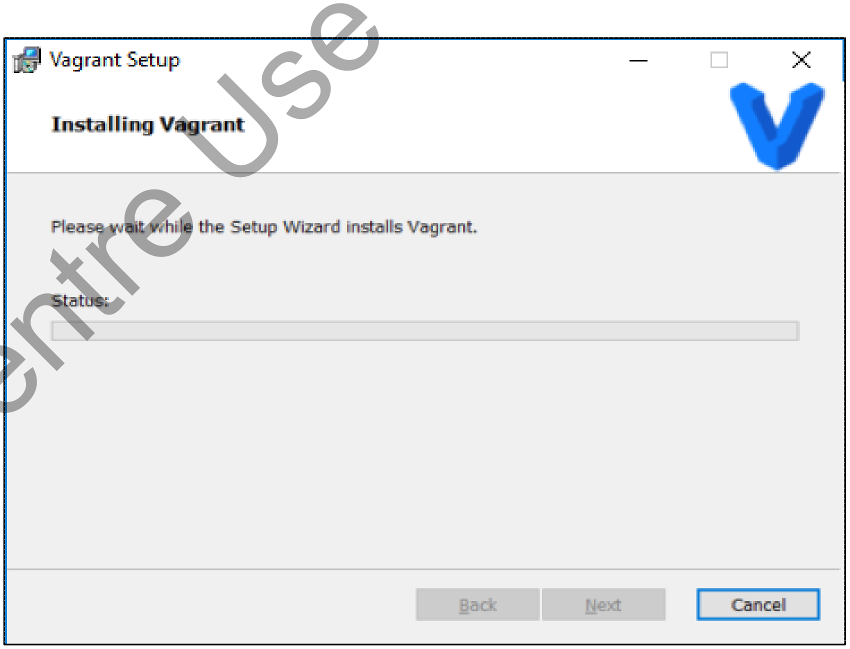
C:\HashiCorp\Vagrant\

# INSTALLING VAGRANT (3-4)


Vagrant Setup

**Ready to install Vagrant**

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

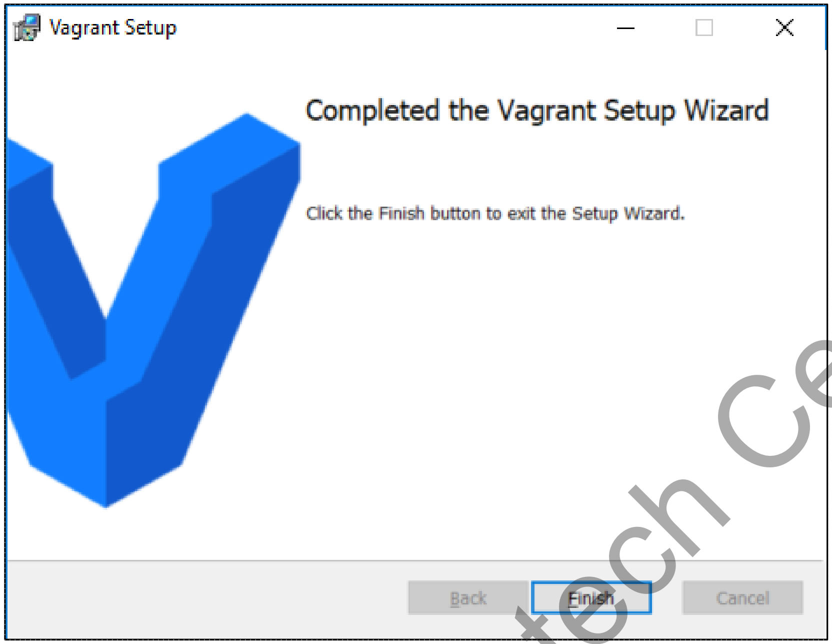

Vagrant Setup

**Installing Vagrant**

Please wait while the Setup Wizard installs Vagrant.

Status:

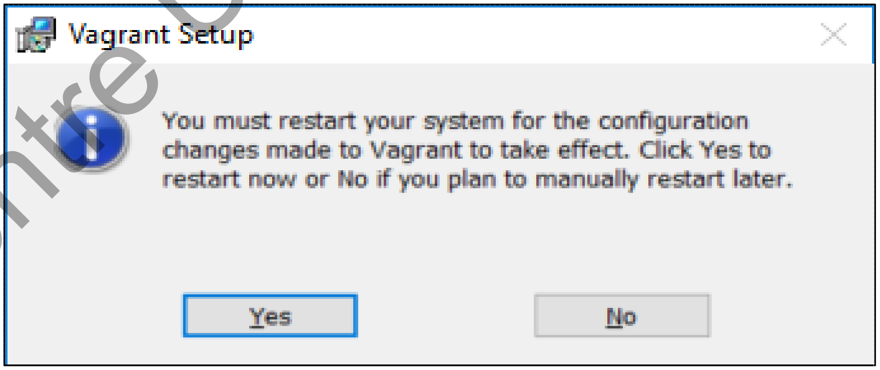
# INSTALLING VAGRANT (4-4)



Vagrant Setup

Completed the Vagrant Setup Wizard

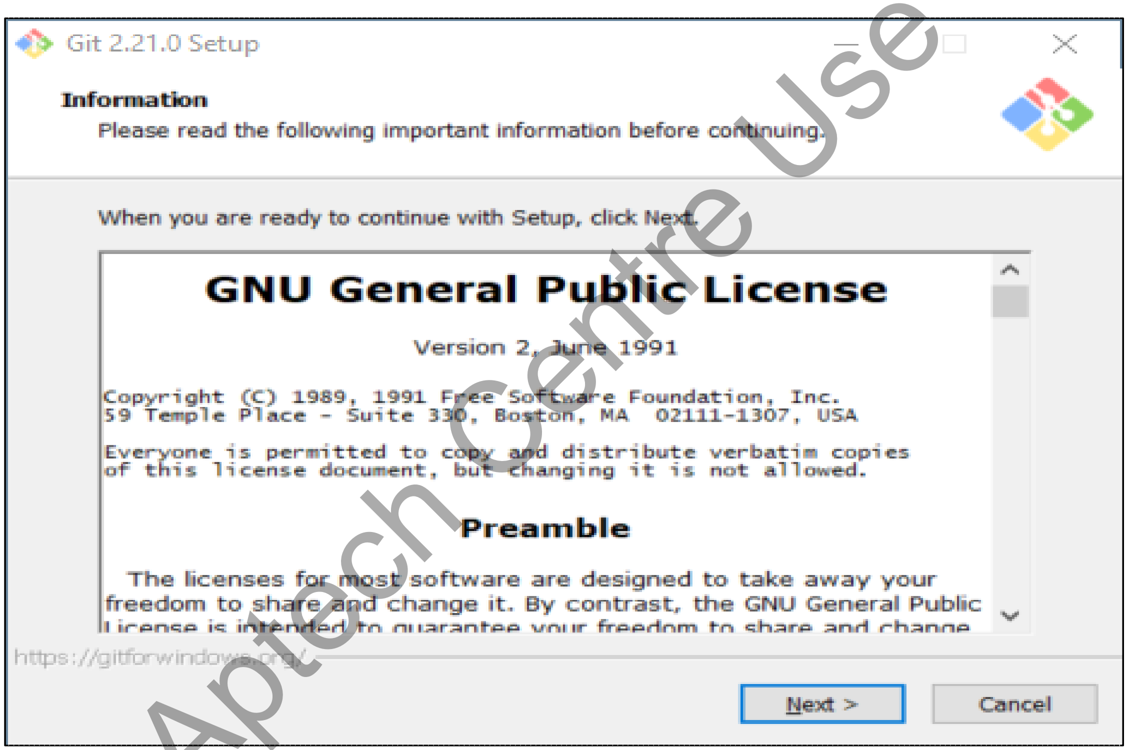
Click the Finish button to exit the Setup Wizard.



Vagrant Setup

You must restart your system for the configuration changes made to Vagrant to take effect. Click Yes to restart now or No if you plan to manually restart later.

# SETTING UP GITBASH (1-7)



**Information**

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

## GNU General Public License

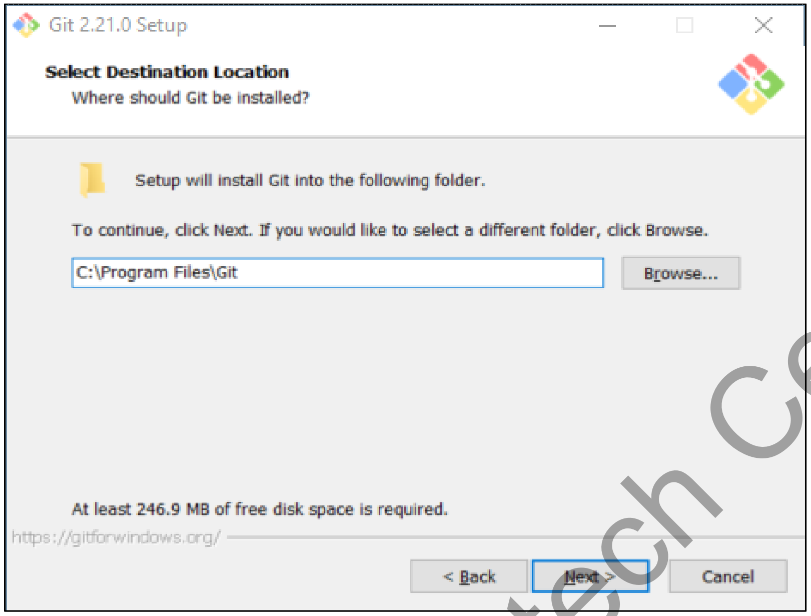
Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
```

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

# SETTING UP GITBASH (2-7)



**Select Destination Location**

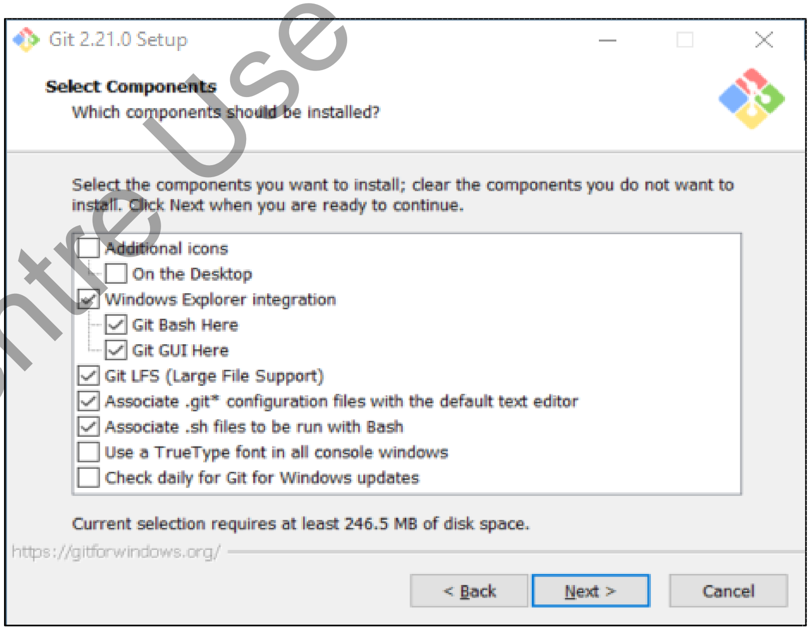
Where should Git be installed?

Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 246.9 MB of free disk space is required.



**Select Components**

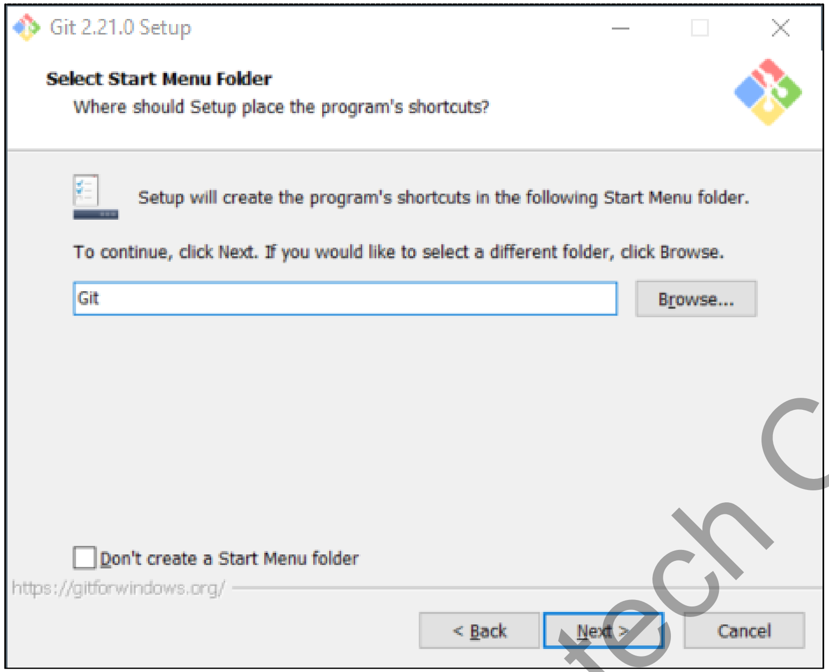
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Git Bash Here
  - ☑ Git GUI Here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Use a TrueType font in all console windows
- □ Check daily for Git for Windows updates

Current selection requires at least 246.5 MB of disk space.

# SETTING UP GITBASH (3-7)



**Select Start Menu Folder**

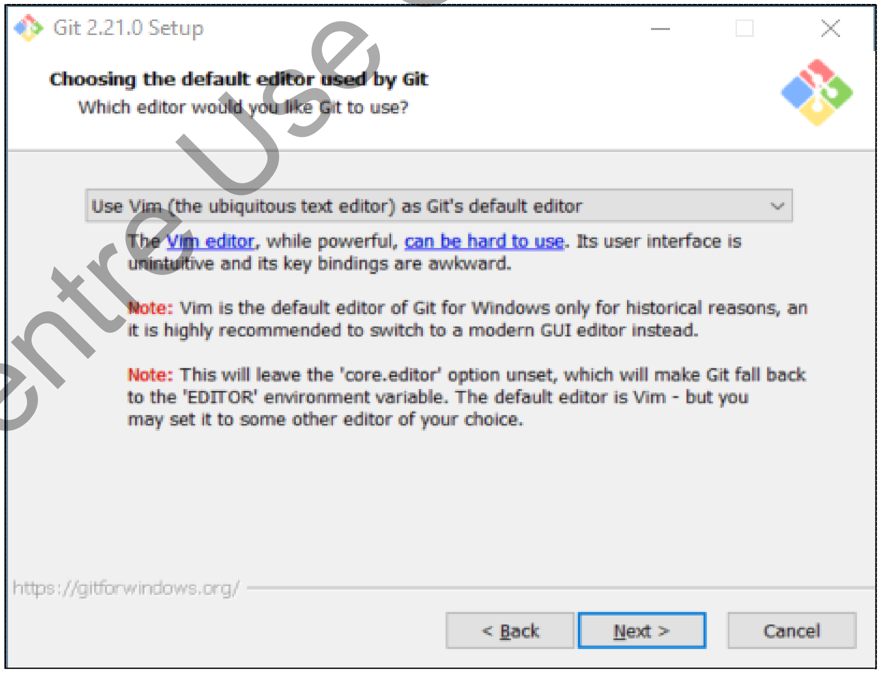
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Git

□ Don't create a Start Menu folder



**Choosing the default editor used by Git**

Which editor would you like Git to use?

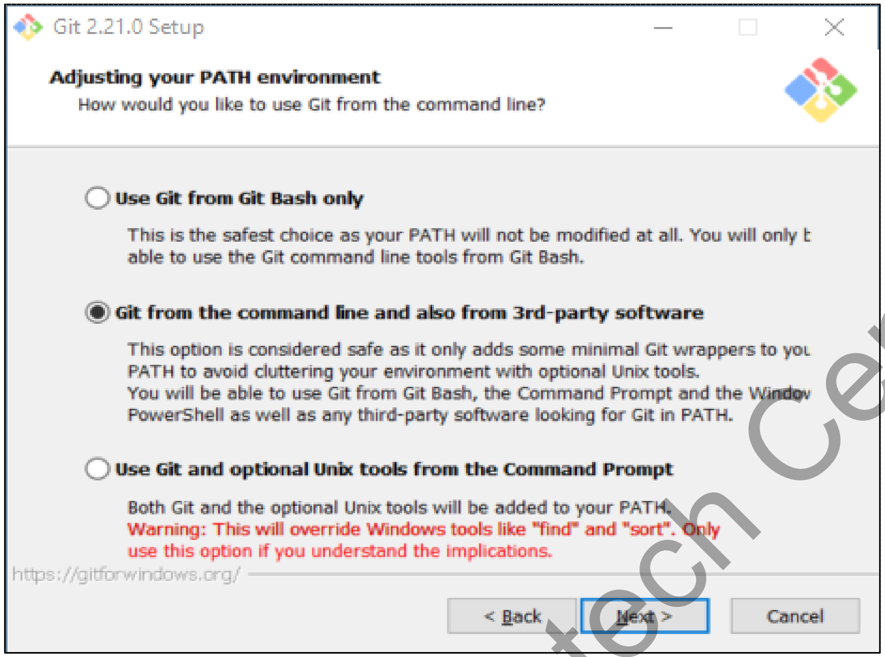
Use Vim (the ubiquitous text editor) as Git's default editor

The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, an it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

# SETTING UP GITBASH (4-7)



Git 2.21.0 Setup

**Adjusting your PATH environment**

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

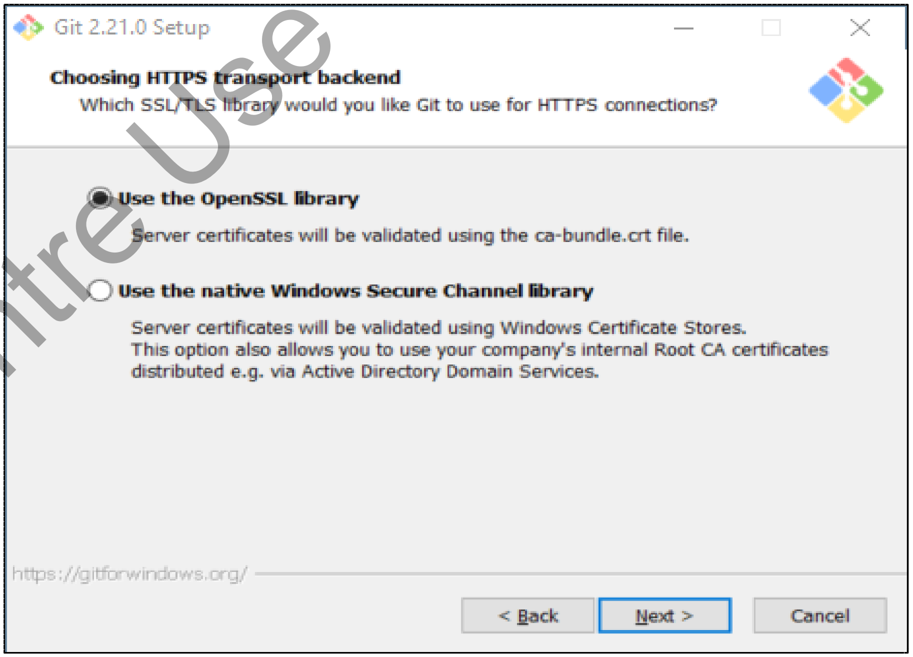
This is the safest choice as your PATH will not be modified at all. You will only b able to use the Git command line tools from Git Bash.

● **Git from the command line and also from 3rd-party software**

This option is considered safe as it only adds some minimal Git wrappers to you PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Window PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.



Git 2.21.0 Setup

**Choosing HTTPS transport backend**

Which SSL/TLS library would you like Git to use for HTTPS connections?

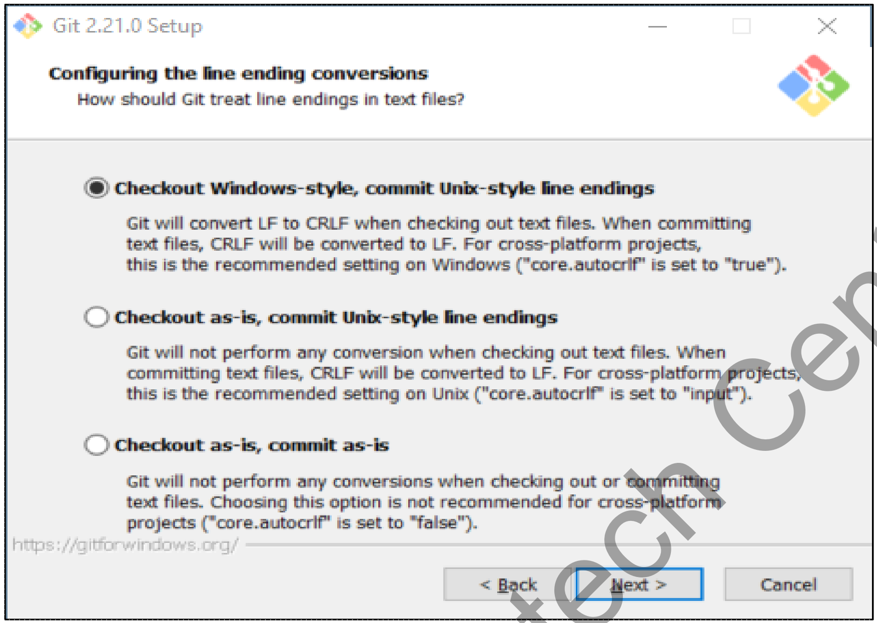
● **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores. This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

# SETTING UP GITBASH (5-7)



**Configuring the line ending conversions**

How should Git treat line endings in text files?

● **Checkout Windows-style, commit Unix-style line endings**

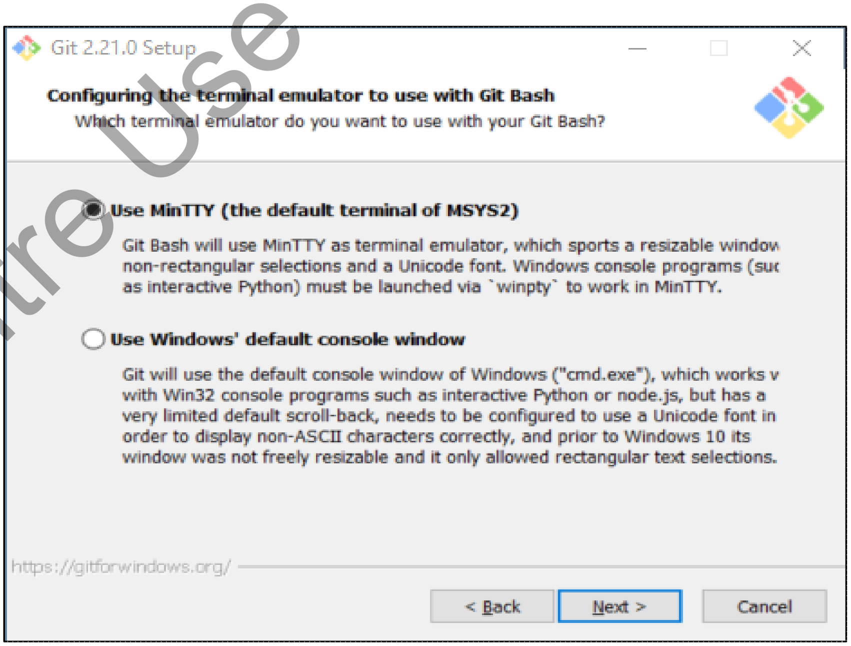
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

○ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").



**Configuring the terminal emulator to use with Git Bash**

Which terminal emulator do you want to use with your Git Bash?

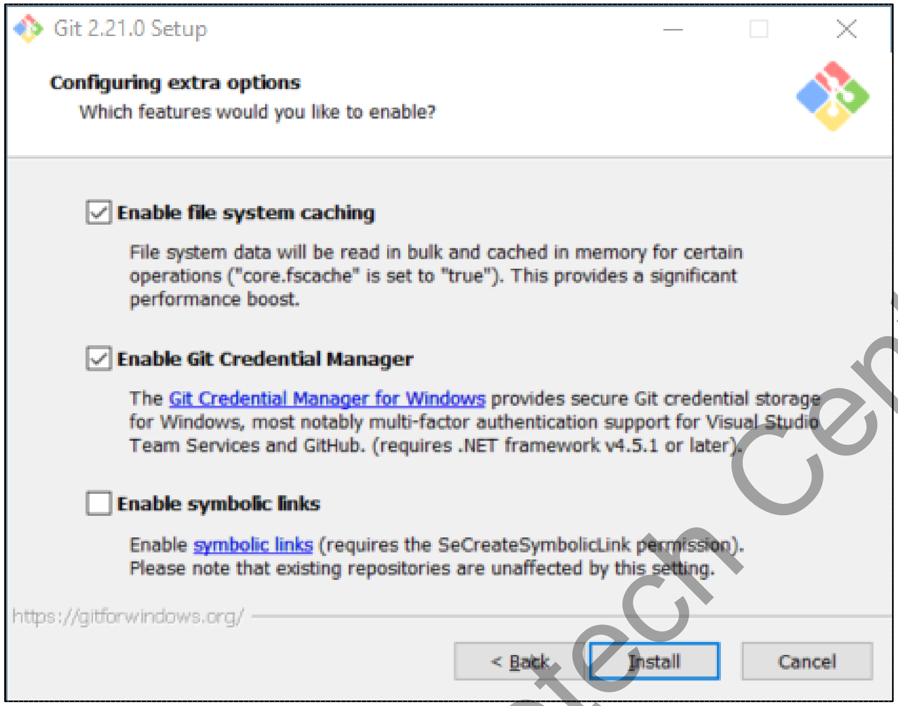
● **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window non-rectangular selections and a Unicode font. Windows console programs (suc as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works v with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

# SETTING UP GITBASH (6-7)



**Configuring extra options**

Which features would you like to enable?

☑ **Enable file system caching**

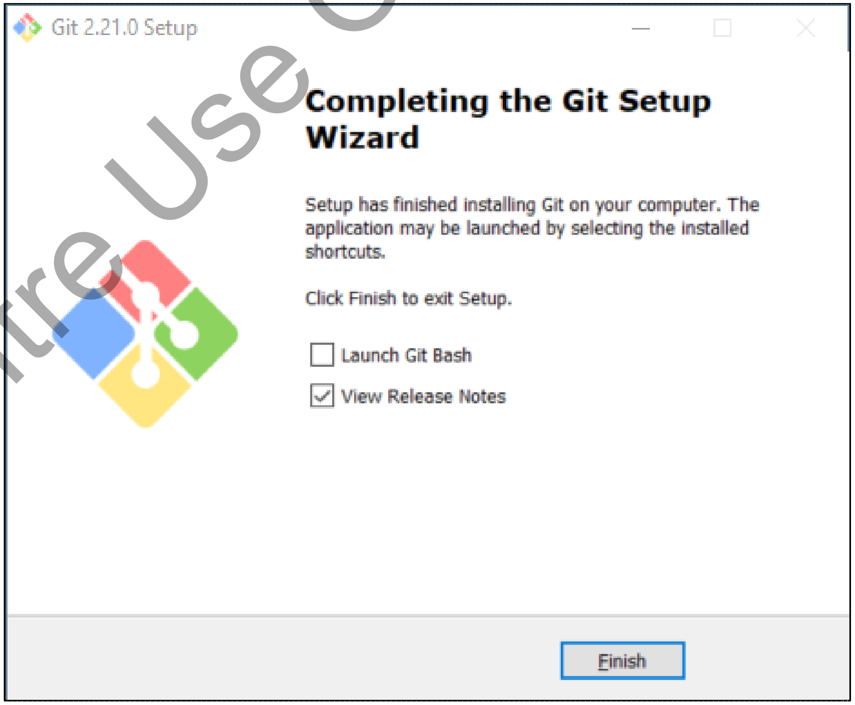
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☑ **Enable Git Credential Manager**

The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or later).

□ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.



**Completing the Git Setup Wizard**

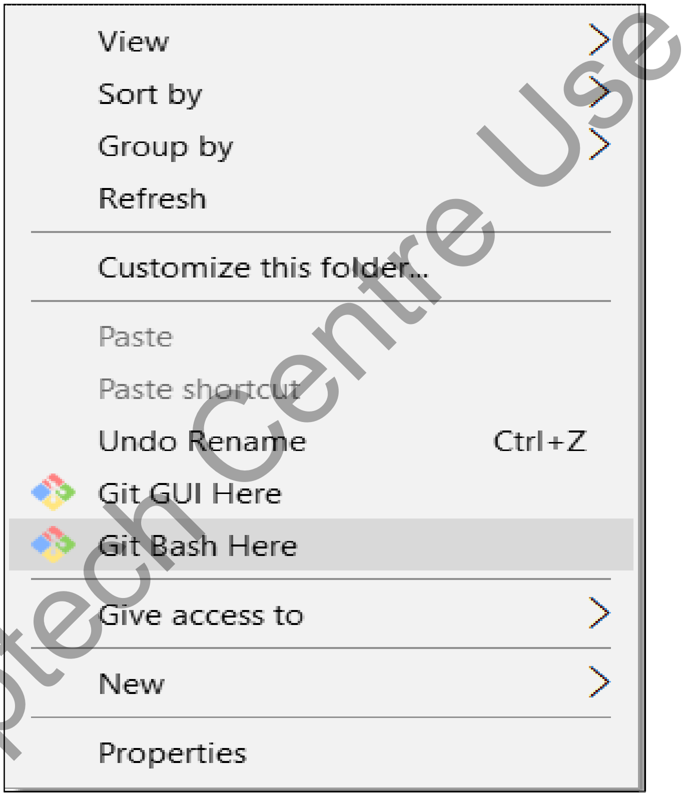
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

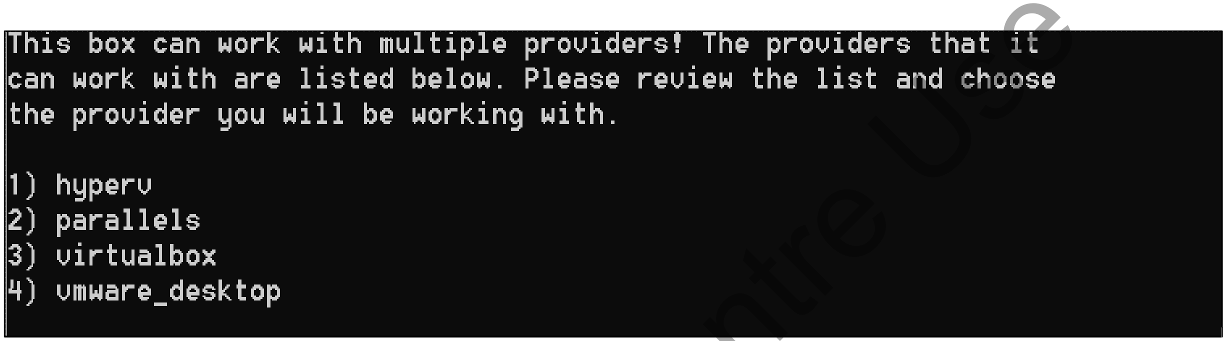
□ Launch Git Bash

☑ View Release Notes

# SETTING UP GITBASH (7-7)



# SETTING UP HOMESTEAD (1-4)

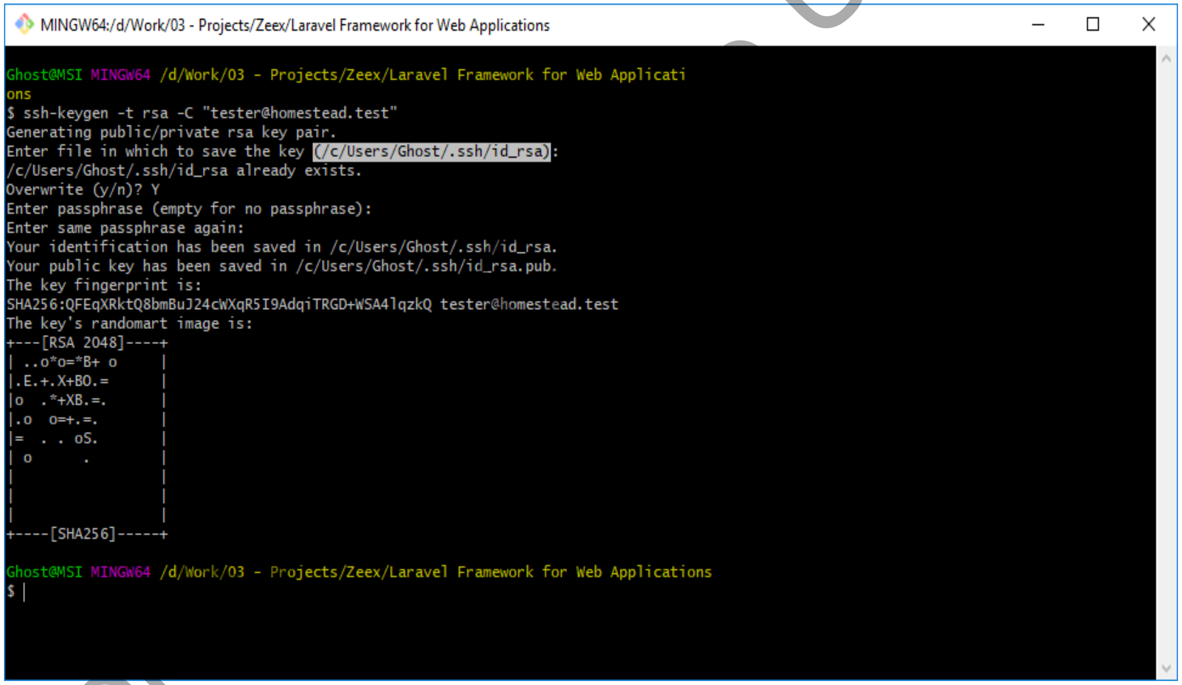


```
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) hyperv
2) parallels
3) virtualbox
4) vmware_desktop
```

```
mkdir homestead
cd homestead
composer require laravel/homestead --dev
.\vendor\bin\homestead make
```

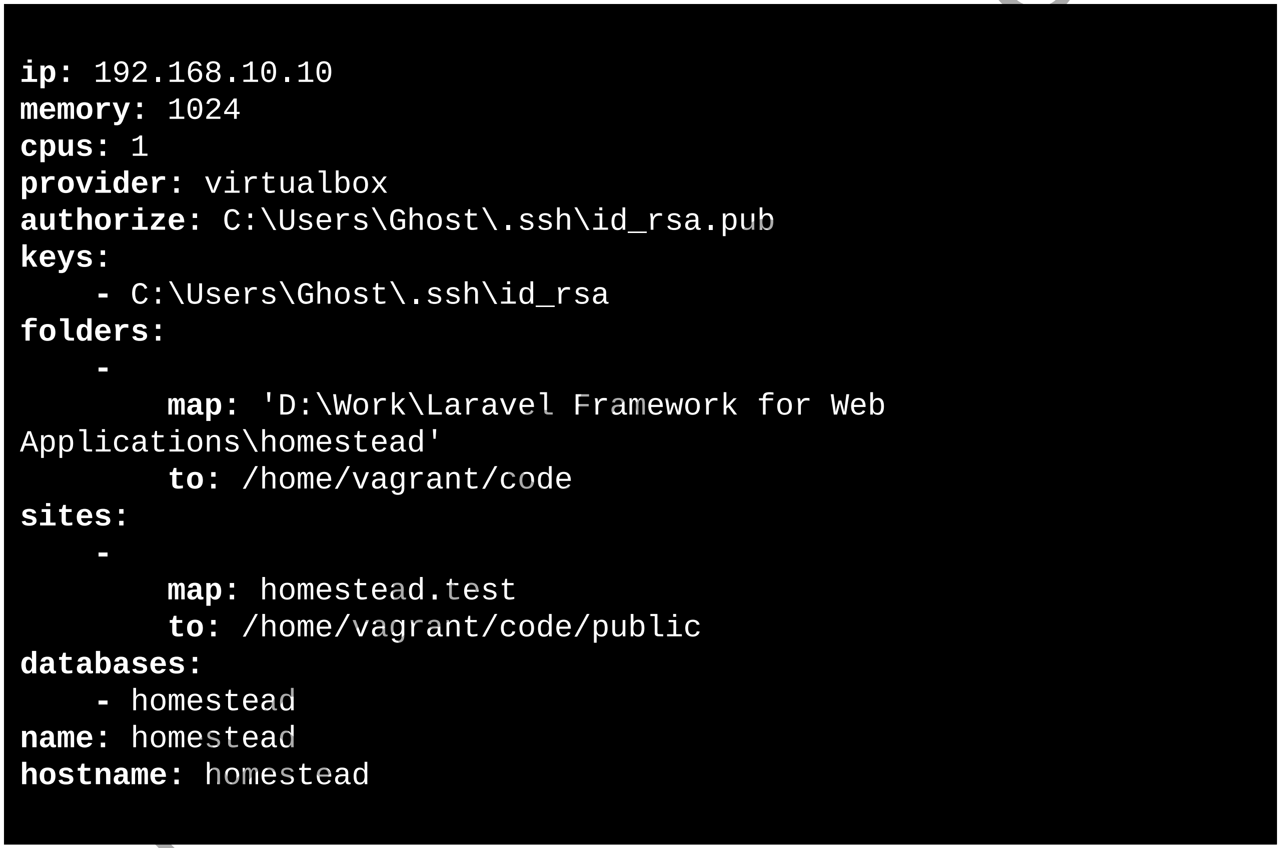
# SETTING UP HOMESTEAD (2-4)

```
ssh-keygen -t rsa -C "tester@homestead.test"
```


MINGW64:/d/Work/03 - Projects/Zeex/Laravel Framework for Web Applications

```
Ghost@MSI MINGW64 /d/Work/03 - Projects/Zeex/Laravel Framework for Web Applicati
ons
$ ssh-keygen -t rsa -C "tester@homestead.test"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Ghost/.ssh/id_rsa):
/c/Users/Ghost/.ssh/id_rsa already exists.
Overwrite (y/n)? Y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Ghost/.ssh/id_rsa.
Your public key has been saved in /c/Users/Ghost/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:QFEqXRktQ8bmBuJ24cWXqR5I9AdqiTRGD+WSA4lqzkQ tester@homestead.test
The key's randomart image is:
+---[RSA 2048]----+
| ..o*o=*B+ o     |
|.E.+.X+BO.=      |
|o  .*+XB.=.      |
|.o  o=+.=.       |
|=  . . oS.       |
| o      .        |
|                 |
|                 |
|                 |
+----[SHA256]-----+

Ghost@MSI MINGW64 /d/Work/03 - Projects/Zeex/Laravel Framework for Web Applications
$ |
```

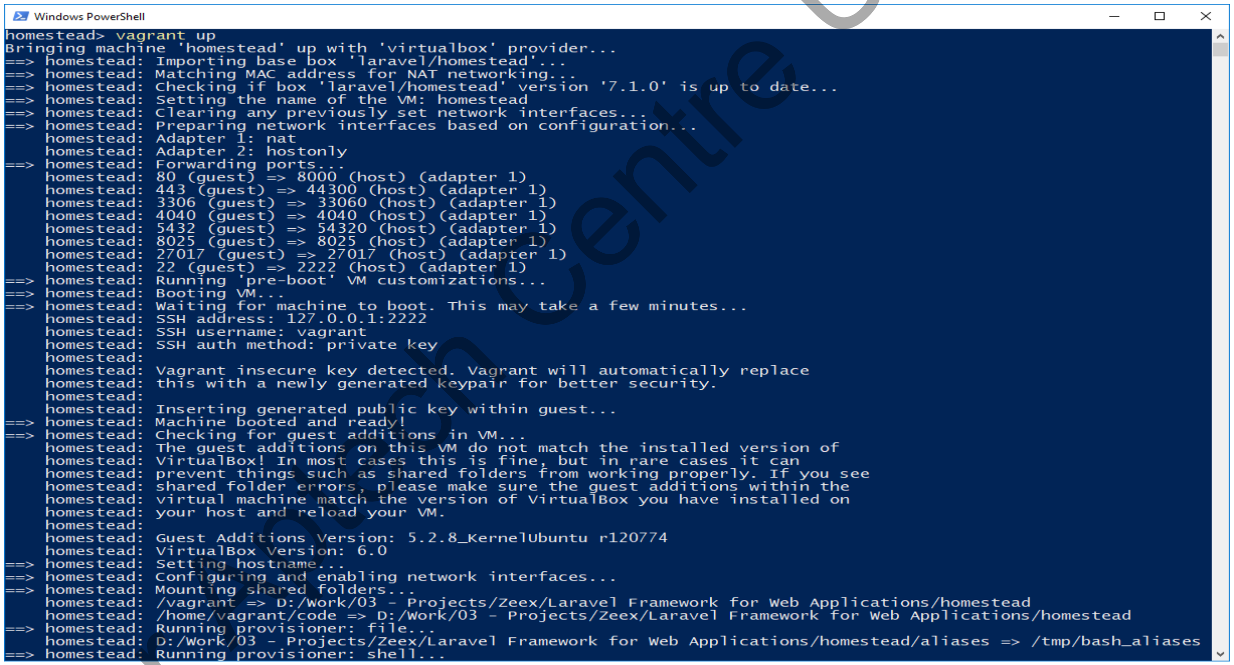
# SETTING UP HOMESTEAD (3-4)



```
ip: 192.168.10.10
memory: 1024
cpus: 1
provider: virtualbox
authorize: C:\Users\Ghost\.ssh\id_rsa.pub
keys:
    - C:\Users\Ghost\.ssh\id_rsa
folders:
    -
        map: 'D:\Work\Laravel Framework for Web
Applications\homestead'
        to: /home/vagrant/code
sites:
    -
        map: homestead.test
        to: /home/vagrant/code/public
databases:
    - homestead
name: homestead
hostname: homestead
```

# SETTING UP HOMESTEAD (4-4)

```
vagrant up
```

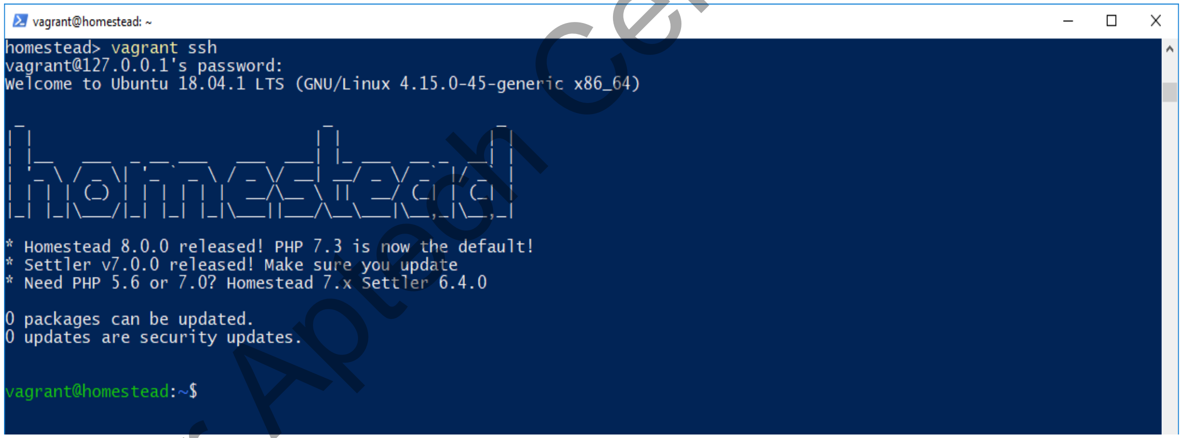


```
homestead> vagrant up
Bringing machine 'homestead' up with 'virtualbox' provider...
==> homestead: Importing base box 'laravel/homestead'...
==> homestead: Matching MAC address for NAT networking...
==> homestead: Checking if box 'laravel/homestead' version '7.1.0' is up to date...
==> homestead: Setting the name of the VM: homestead
==> homestead: Clearing any previously set network interfaces...
==> homestead: Preparing network interfaces based on configuration...
    homestead: Adapter 1: nat
    homestead: Adapter 2: hostonly
==> homestead: Forwarding ports...
    homestead: 80 (guest) => 8000 (host) (adapter 1)
    homestead: 443 (guest) => 44300 (host) (adapter 1)
    homestead: 3306 (guest) => 33060 (host) (adapter 1)
    homestead: 4040 (guest) => 4040 (host) (adapter 1)
    homestead: 5432 (guest) => 54320 (host) (adapter 1)
    homestead: 8025 (guest) => 8025 (host) (adapter 1)
    homestead: 27017 (guest) => 27017 (host) (adapter 1)
    homestead: 22 (guest) => 2222 (host) (adapter 1)
==> homestead: Running 'pre-boot' VM customizations...
==> homestead: Booting VM...
==> homestead: Waiting for machine to boot. This may take a few minutes...
    homestead: SSH address: 127.0.0.1:2222
    homestead: SSH username: vagrant
    homestead: SSH auth method: private key
    homestead:
    homestead: Vagrant insecure key detected. Vagrant will automatically replace
    homestead: this with a newly generated keypair for better security.
    homestead:
    homestead: Inserting generated public key within guest...
==> homestead: Machine booted and ready!
==> homestead: Checking for guest additions in VM...
    homestead: The guest additions on this VM do not match the installed version of
    homestead: VirtualBox! In most cases this is fine, but in rare cases it can
    homestead: prevent things such as shared folders from working properly. If you see
    homestead: shared folder errors, please make sure the guest additions within the
    homestead: virtual machine match the version of VirtualBox you have installed on
    homestead: your host and reload your VM.
    homestead:
    homestead: Guest Additions Version: 5.2.8_KernelUbuntu r120774
    homestead: VirtualBox Version: 6.0
==> homestead: Setting hostname...
==> homestead: Configuring and enabling network interfaces...
==> homestead: Mounting shared folders...
    homestead: /vagrant => D:/Work/03 - Projects/Zeex/Laravel Framework for Web Applications/homestead
    homestead: /home/vagrant/code => D:/Work/03 - Projects/Zeex/Laravel Framework for Web Applications/homestead
==> homestead: Running provisioner: file...
    homestead: D:/Work/03 - Projects/Zeex/Laravel Framework for Web Applications/homestead/aliases => /tmp/bash_aliases
==> homestead: Running provisioner: shell...
```

# SETTING UP LARAVEL/VERIFYING INSTALLATION

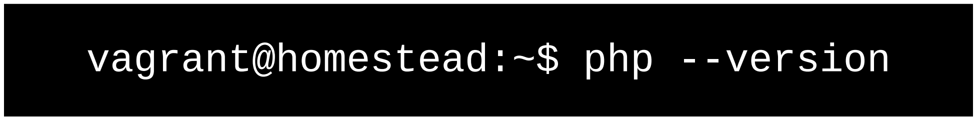
```
laravel new <project-name>
```

```
vagrant ssh
```

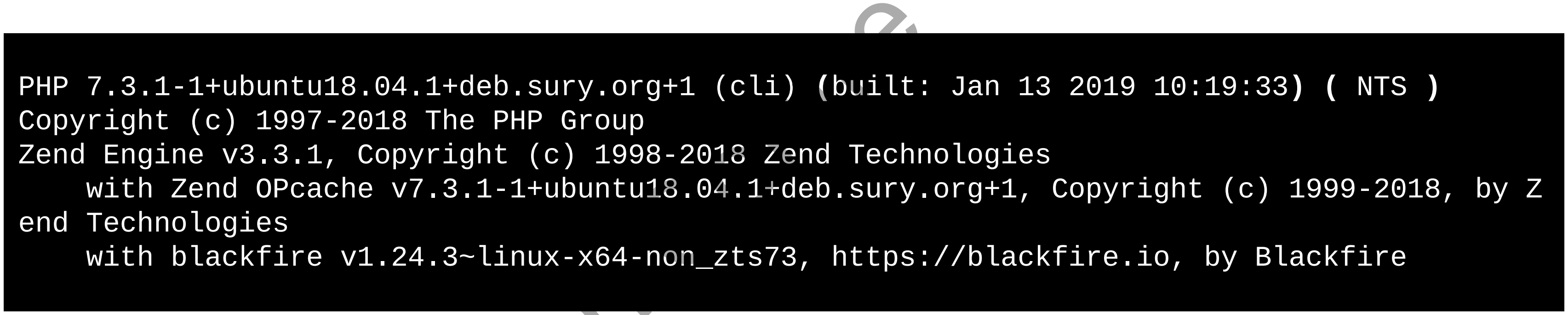


```
homestead> vagrant ssh
vagrant@127.0.0.1's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 _                               _                 _
| |                             | |               | |
| |__   ___  _ __ ___   ___  ___| |_ ___  __ _  __| |
| '_ \ / _ \| '_ ` _ \ / _ \/ __| __/ _ \/ _` |/ _` |
| | | | (_) | | | | | |  __/\__ \ ||  __/ (_| | (_| |
|_| |_|\___/|_| |_| |_|\___||___/\__\___|\__,_|\__,_|

* Homestead 8.0.0 released! PHP 7.3 is now the default!
* Settler v7.0.0 released! Make sure you update
* Need PHP 5.6 or 7.0? Homestead 7.x Settler 6.4.0

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


vagrant@homestead:~$
```

# VERIFYING VERSIONS (1-2)



```
vagrant@homestead:~$ php --version
```



```
PHP 7.3.1-1+ubuntu18.04.1+deb.sury.org+1 (cli) (built: Jan 13 2019 10:19:33) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.1, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.1-1+ubuntu18.04.1+deb.sury.org+1, Copyright (c) 1999-2018, by Z
end Technologies
    with blackfire v1.24.3~linux-x64-non_zts73, https://blackfire.io, by Blackfire
```

# VERIFYING VERSIONS (2-2)

```
vagrant@homestead:~$ composer -V
```

```
Composer version 1.8.4 2019-02-11 10:52:10
```

```
vagrant@homestead:~$ sqlite3 -version
```

```
3.22.0 2018-01-22 18:45:57 0c55d179733b46d8d0ba4
d88e01a25e10677046ee3da1d5b1581e86726f2alt1
```

# SUMMARY

- Components in an environment to develop Web applications with Laravel include PHP, Composer, Virtual Box, Vagrant, Git Bash, and Homestead.
- Composer is a dependency management tool for PHP.
- VirtualBox is a free and open source hypervisor that will allow running guest VMs, such as the Homestead environment.
- Vagrant is an automation tool for building and managing virtual machine environments. Homestead is a vagrant box.
- Git Bash is a shell that comes along with Git for Windows. It helps to use git commands and provides Unix-type commands that can be run on Windows.
- Homestead is a virtual environment that provides resources to efficiently code Web applications in Laravel using PHP.
- After installing the components, verify the installation and check version number by sshing into the newly deployed Homestead environment.