

# "HELLO" ENVIRONMENTS



**Learn Modern  
Development Environments**

# "Hello" Environments

## Learn Modern Development Environments

Rick Hernandez

This book is for sale at <http://leanpub.com/hello-dev-environments>

This version was published on 2015-12-25



\* \* \* \* \*

This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

\* \* \* \* \*

© 2015 Rick Hernandez

# Table of Contents

## Introduction

[About the author](#)

[Who this book is for?](#)

[Prerequisites before getting started](#)

[Code Samples](#)

[Book Conventions](#)

[What to expect out of the book?](#)

## Development Environments

[The Basics](#)

[Local Environment](#)

[Virtual Environment](#)

[Cloud Environment](#)

## Text Editors

[Vi / VIM](#)

[Atom by Github](#)

[Sublime Text](#)

[Brackets by Adobe](#)

## Command Line Tools

[OSX](#)

[Linux](#)

[Windows](#)

## Congratulations!

# Introduction

I want to take this time to say **thank you** and congratulate you for taking the first step to creating the perfect development environment. This book contains solid techniques on how to setup your ideal development environment. Throughout this book, you will learn the in's and out's of multiple development environments, text editors (Vi, Vim, Atom, Sublime, Brackets), understand how to work with local, virtual and private clouds (VirtualBox, Vagrant, Cloud9), you will also learn how to automate the most common task using the command line. But most importantly you will have a good understanding of what to look for when you are creating your "*ideal*" development environment.

## About the author



Rick H

Rick Hernandez works as a software developer with small to mid-size companies. He has been responsible for evaluating business requests to determine feasibility, identifying options, and recommend solutions for software development enhancements. He has assisted with interpreting customer requirements into conceptual design specifications, and has developed interfaces/prototypes and maintained software solutions. He also has written over 200 technical publications published at [Code With Intent](#). Rick has a deep passion and curiosity to help others succeed.

## Say Hello

My intention with this book is to get you up and running with your ideal development environment. If for whatever reason you might have a question, comment or suggestion about the book, I encourage you to take the time and write into [Code With Intent](#).

## Who this book is for?

The purpose of this book is to help you learn to create your ideal development environment. In the process of this learning you will have to explore different

topics and be willing to look into different editors and not just be stuck with your 1976 text editor.

If you have ever had any of the following questions and did not know how to answer them, then this book is for you.

*“How can I setup my development environment for programming with JavaScript?”*

*“What’s all the buzz about Vagrant?”*

*“How can I get my entire team setup using the same development environment?”*

*“What’s the best web browser?”*

*“Why don’t JavaScript developer show any love for Windows Developers?”*

*“What is the difference between Cloud Environment and a Virtual Environment?”*

*“I don’t want to use Linux! Can you show me Windows?”*

By the end of the book you will be able to answer all of this questions.

I assure you that if you have perseverance, dedication, and creativity to finishing this book, you will be off to having an awesome development environment that all your coworkers will be jealous of.

## **Prerequisites before getting started**

Before getting started with this book, you need to have a clear understanding of the following concepts to enable you to be successful with this book.

- Basic command line experience

If for what ever reason you are not completely comfortable with the above requirements, I have provided some resources for you to get you going before getting started with this book.

- [Bash Basics](#)
- [The UNIX Operating System](#)

Software and Hardware are constantly evolving this book uses the following operating system's as a baseline.

## Software

- OSX 10.10 or higher
- Ubuntu 12.04 or higher
- Windows 7 or higher
- Modern Browser (Firefox, Chrome, Safari)

## Hardware

- Minimum of 2GB of free memory
- 4GB of free hard drive space (optional)

This book will only focus on the mentioned operating systems, if you are running a earlier version of one the mentioned operating systems, this guide will need to be modified to work with that operating system. I would recommend you to just read the abstracts of every chapter and get an idea for what you need and then move on.

## Code Samples

The book contains a variety of code samples and snippets that are used to illustrate a specific part of the book. None of the code builds upon each other, feel free to fork any part of the code and use it for your own setup.

All of the snippets used throughout the book are released under the [MIT License](#).

The complete list of the snippets can be found on [github](#).

## Book Conventions



### Pro Tip

Pro Tips contain information that is well established within the professional development group of programmers.



### Suggestion

A suggestion is an opinionated or suggestive comment toward a specific topic.



### Warning

Warnings are well established problems or common errors.

## Code Examples

```
1 sudo apt-get update  
2 sudo apt-get upgrade
```

## What to expect out of the book?

This book is used as an educational mean, to convey one main message.

### Setting up the perfect Development Environment

This book is **NOT** here to teach you how to use this tools, some of this tools take years to master. Each of the tools that I cover, have a wealth of resources where you can learn more about them.

*I will not be covering shortcuts, the command line, system administration, or macros.* This book is to help you understand what tools you need to make you a more productive developer, by helping you automate some of the systems that help you create software.



### Pro Tip

**Remember:** The key to being a **GREAT** programmer is being able to learn new skills.

## Questions and Answers

---

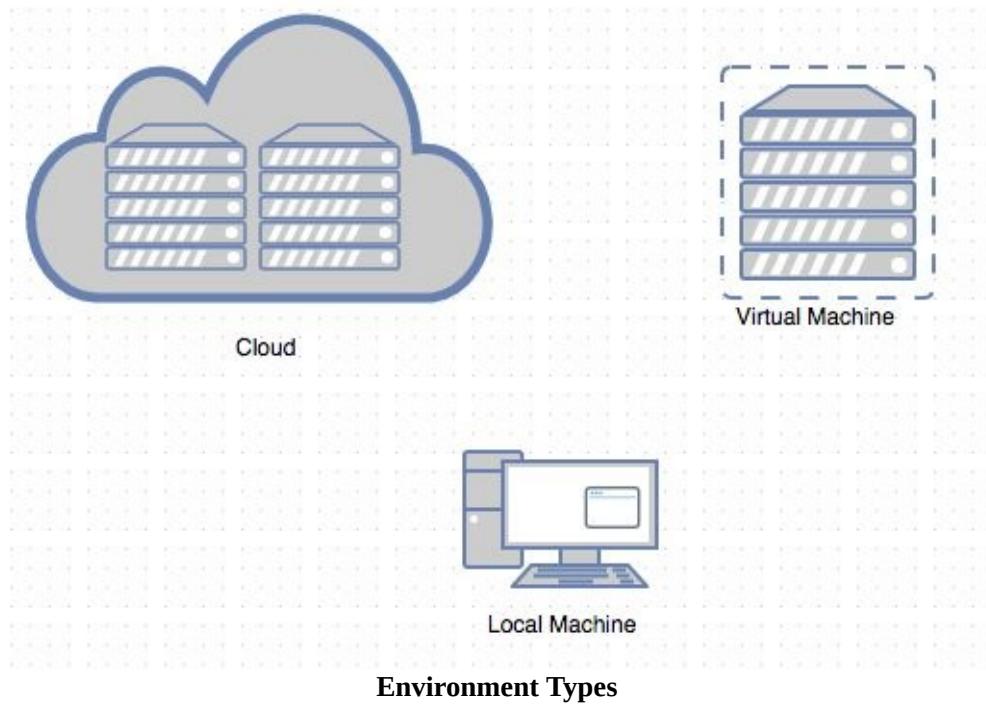
**Is this book language agnostic? No,** this book was design with one specific developer in mind, a JavaScript developer. If you are using another language such as C++, Java, Go, Perl etc this book is not for you.

**Will you cover IDE's, for example Netbeans, AWebStorm etc?** Nope.

**Does this book have a bias towards an operating system?** Yes! This book does have a bias for **\*nix** systems. Why? for me is due to my background in Computer Science and the bias from my instructors at my University, for others it could be a number of different reasons.

**Will you cover how to use the command line?** Nope.

# Development Environments



The development environment that you decide to work on will dictate a large portion of your overall happiness as a programmer. Choosing the right platform and the right text editor could be one of the reasons why you choose to keep working as a programmer or decide to move on to the next great thing. The right choice will vary with everyone's circumstances as it is based on your current needs and budget.

**What is a Development Environment?** Computer programmers require a set of specific tools when writing a software application. Your development environment is used to create software.

In this chapter you will be learning about the three different types of development environments that are the most common when working with JavaScript.

1. Local Environment
2. Virtual Environment

### 3. Cloud Environment

Remember depending on your specific circumstances, you will tend to lean towards a specific setup. Take for example the local environment, if you just want to get started with programming then this is the fastest way to do so.

If you are working with a group of teammates, you might have to look into a virtual or cloud environment. Now power up your desktop, or laptop and let's get started setting up your development environment.

## The Basics

**What is a Local Environment?**: Typically, this will be the environment that is included with your operating system.

**What is a Virtual Environment?**: This goes one step further than the local environment by installing a piece of software on the local environment to be able to virtualize instances of a machine onto the local machine.

**What is a Cloud Environment?**: This goes even further than the virtual environment, by completely eliminating the hardware and just provides you access to the machine over an SSH or internet connection in order for you to configure your specific environment.

**What is the difference between Cloud Environment and a Virtual Environment?** Virtual and cloud environments are completely different concepts although cloud environments typically use some type of virtualization to provide you with a private environment.

**What do Local Environments, Virtual Environments, and Cloud Environments all have in common?** **Environment Variables**, which are a set of dynamic values that affect the way a program will behave on your computer. All of these variables are configured and managed by your operating system. The main reason why programmers use them is to be able to have a dynamic value that is able to be referenced.

Below you will find the list of the most common environment variables along with the location of where you can configure these variable values.

<b>Variable Description</b>	
<b>Linux</b>	
<b>Variable</b>	<b>Description</b>
\$PATH	Semicolon delimited list of directories
\$HOME	The path to the home directory

<b>Windows</b>	
<b>Variable</b>	<b>Description</b>
%PATH%	Semicolon delimited list of directories
%HOMEDIR%	The path to the home directory



### Pro Tip

**Remember:** Environment variables can have scope and a value, just like any other variable.

## Variable Location

Depending on what operating system that you are in the location of this variables will differ. The following locations are for global variables.

**OSX:** The default location for custom environmental variables in OSX the .bash\_profile file.

.bash\_profile

---

1 \$ ~/.bash\_profile

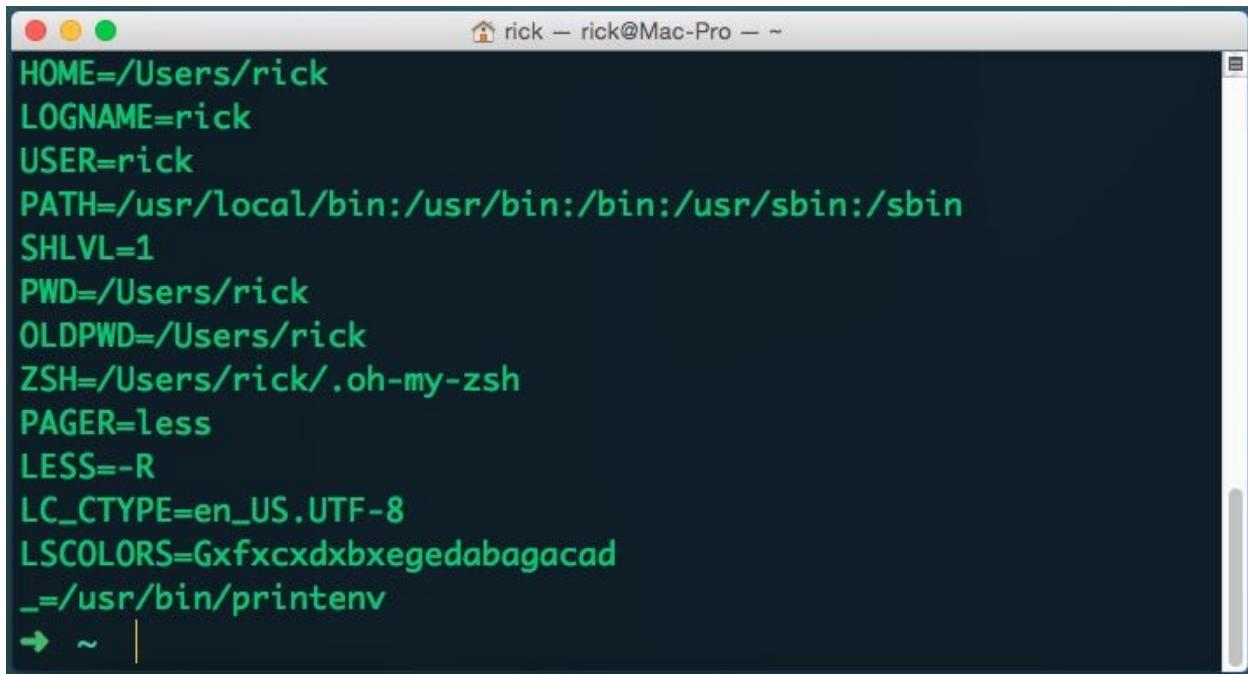
---

To list all the variables that are being used by your system use the printenv command.

.printenv

---

```
1 $ ~ printenv
```



A screenshot of a macOS terminal window titled "rick" with the identifier "rick@Mac-Pro". The window shows the output of the "printenv" command. The output lists various environment variables with their values. The variables include HOME=/Users/rick, LOGNAME=rick, USER=rick, PATH=/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin, SHLVL=1, PWD=/Users/rick, OLDPWD=/Users/rick, ZSH=/Users/rick/.oh-my-zsh, PAGER=less, LESS=-R, LC\_CTYPE=en\_US.UTF-8, LSCOLORS=Gxfxcxdxbxgedabagacad, \_=/usr/bin/printenv, and the prompt ~ |.

printenv

**Linux:** The default location for custom environmental variables in Linux is the `.bash_profile` file.

.bash\_profile

```
1 ~/ .bash_profile
```

To list all the variables that are being used by your system use the `printenv` command.

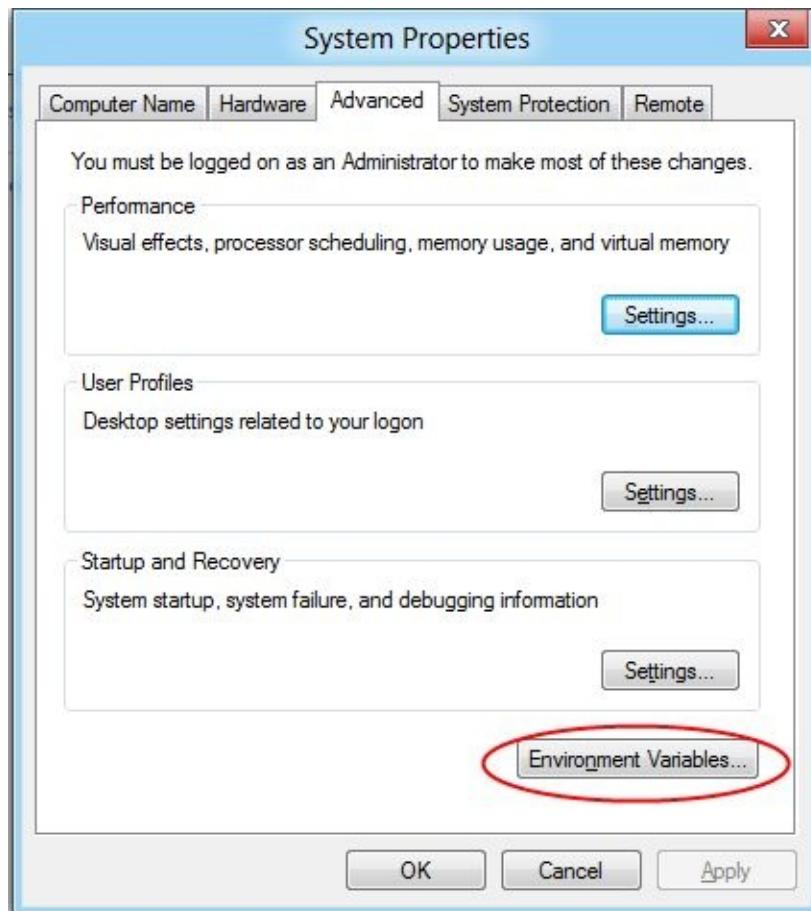
.printenv

```
1 $ ~ printenv
```

```
rick — rick@ubuntu: ~ ~ ~
SSH_CLIENT=192.168.56.1 51335 22
SSH_CONNECTION=192.168.56.1 51335 192.168.56.101 22
SSH_TTY=/dev/pts/0
TERM=xterm-256color
XDG_SESSION_ID=1
XDG_RUNTIME_DIR=/run/user/1000
SHLVL=1
PWD=/home/rick
OLDPWD=/var/www/hello_text_editors/en
ZSH=/home/rick/.oh-my-zsh
PAGER=less
LESS=-R
LSCOLORS=Gxfxcxdxbxegedabagacad
_=~/usr/bin/printenv
→ ~
```

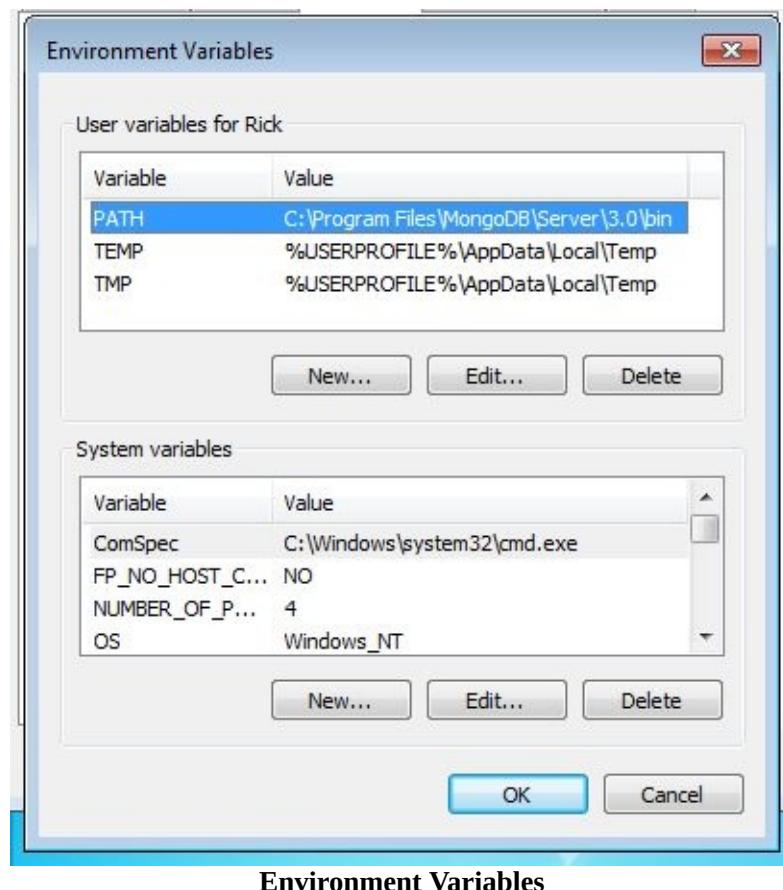
printenv

**Windows:** In Windows 7 you can find the location path for the variables in the following location.



## Windows Location

1. Click the Start button and right-click the computer option in the start menu, and select Properties.
2. Click the Advanced System Settings link in the left column.
3. In the System Properties window, click on the Advanced tab, then click the Environment Variables button near the bottom of the window (pictured above).
4. You find the Environment Variables window (pictured below)



## Variable Assignment

When writing your application you might want to set or read an environment variable. One of the most common custom variables used in *ExpressJS* is the `NODE_ENV` variable. This is a custom variable that is used to distinguish between *production* and *development* environments.

## OSX

**Setting a variable:** To set the variable use the command line.

NODE\_ENV

---

```
1 $ export NODE_ENV=development
```

---

**Reading a variable:** To read the variable using the command line.

NODE\_ENV

---

```
1 $ echo NODE_ENV
```

---

**Reading a variable:** To read the variable using JavaScript.

JavaScript

---

```
1 console.log(process.env.NODE_ENV);
```

---

**Linux**

**Setting a variable:** To set the variable use the command line.

NODE\_ENV

---

```
1 $ export NODE_ENV=development
```

---

**Reading a variable:** To read the variable using the command line.

NODE\_ENV

---

```
1 $ echo NODE_ENV
```

---

**Reading a variable:** To read the variable using JavaScript.

JavaScript

---

```
1 console.log(process.env.NODE_ENV);
```

---

**Windows**

**Setting a variable:** To set the variable use the command line.

NODE\_ENV

---

```
1 set NODE_ENV=production
```

---

**Reading a variable:** To read the variable using the command line.

NODE\_ENV

---

```
1 ECHO %NODE_ENV%
```

---

**Reading a variable:** To read the variable using JavaScript.

JavaScript

```
1 console.log(process.env.NODE_ENV);
```



### Warning

Environment variables optionally can have the value of `undefined`, always check the variable before using it in your program.

## Summary

Remember it's your personal development environment, it needs to be as unique as you. You will be using it hours on end, if you decide to use windows great, if you go with linux great, or OSX then great.

## Local Environment



Local Machine

Local Machine

One of the great things about working with JavaScript, is that you don't need to have a complex development environment. Most modern computers have a web browser and a pre installed text editor which is all you need to start to program using JavaScript. In this section we will cover how to create a simple program with this tools.

## Modern Web browsers

Most modern operating systems support the ability for you to install and setup a browser. If you are running windows then IE is pre installed, if you running Linux you might get Chrome or Midori for OSX users you will have Safari pre installed.

The rest of this book will focus on using Chrome. If your computer does not have chrome installed you can download the latest version below.

### [Chrome Download](#)

**Hold on!** What about the rest of the browsers? Great question! this is one of the many decisions that you must take in order for you to create the “*Perfect development environment*”. Here is a list of the most popular web browsers for 2016.

1. Firefox
2. Chrome
3. Opera
4. Safari
5. Internet Explorer
6. Torch
7. Midori



#### Pro Tip

If you decide to be a web developer in the future you will have to support this list of browsers, and many more.

## The JavaScript console

The JavaScript console provides you the ability to log messages, enter executable code or evaluate regular JavaScript expressions.

**How to open the console?** The JavaScript Console can be accessed in three different places.

## 1. Use the keyboard

1. Mac    Command + Option + J (Mac)

2. Windows and Linux    Control + Shift + J (Windows/Linux).

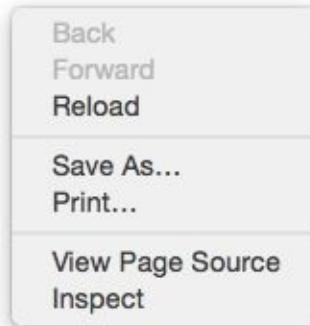
## 2. Use the Menu

1. Select  > More Tools > Developer Tools.

**Menu**

## 3. Use the Mouse

1. Right click anywhere inside a webpage and select inspect



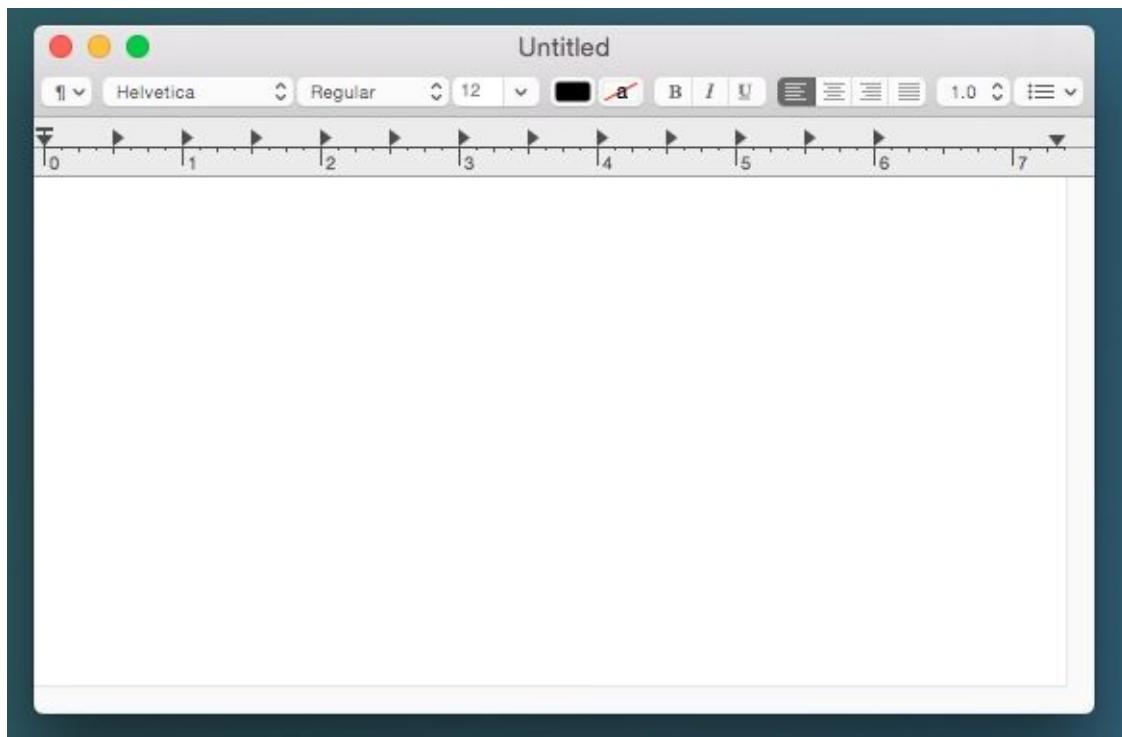
**Menu Page**

Now that you know how to open the console, this will be used later when creating your first JavaScript program.

## The Text Editors

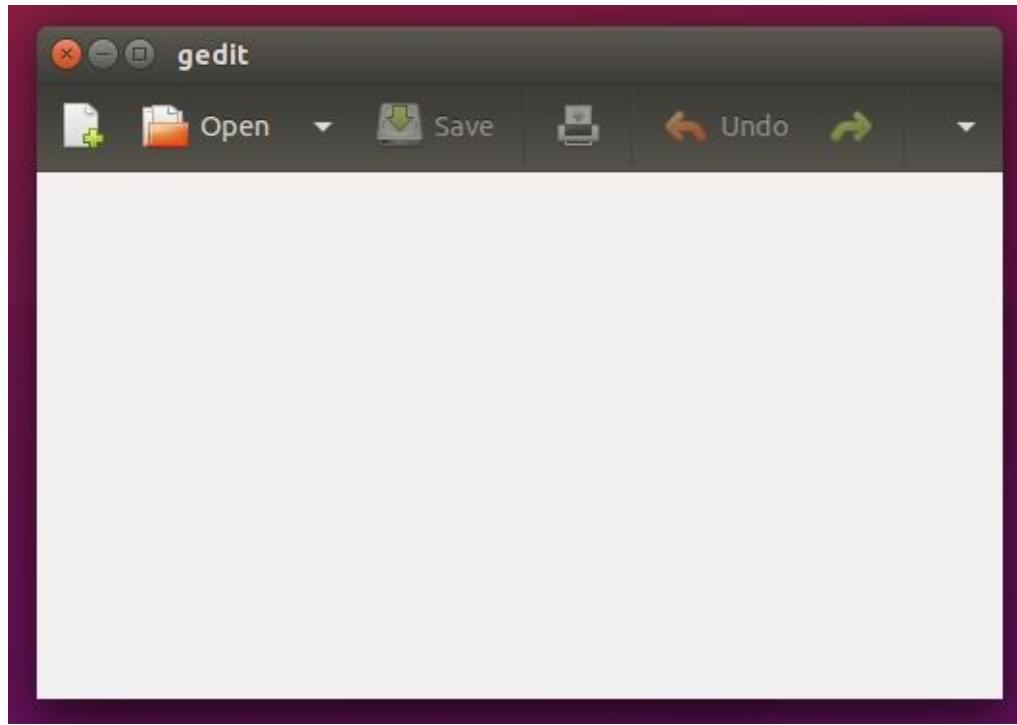
In your local environment, the operating system will provide you with a text editor that is pre installed. Below you will find the list of the most common text editors that are included in your operating system.

**OSX:** In OSX you will have an application named `TextEdit`



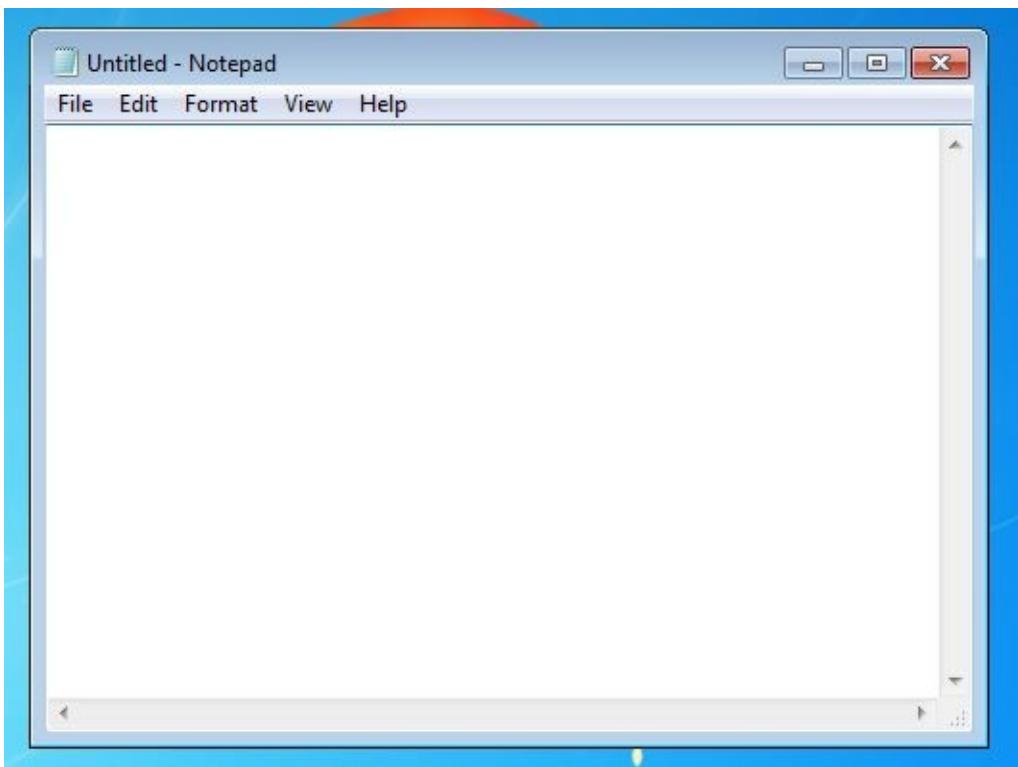
**Text Edit**

**Linux:** In Ubuntu you will have an application named gEdit



**G Edit**

**Windows:** In Windows 7 you will have an application named Notepad



Note Pad

**Can I use Pages, LibreOffice or Microsoft Word to write code?:** Yes and No, although each of this applications allow you to use them as text editors, their main purpose is to be used as a word processor not a text editor. The difference is that word processor will add extra information to the files that are saved, typically meta data such as formatting, font colors, font types etc. Your JavaScript programs won't require any of this. This is another decision that you must make to find your "*Perfect development environment*".

## Running a simple program

Your browser requires an HTML page in order for it to be rendered inside the browser. JavaScript requires to be inside an HTML page in order for the browser to interpret the JavaScript Code.

hello.html

```
1 <script type="text/javascript" src="hello.js" async></script>
```

hello.js

```
1 console.log('Hello World');
```

Now just double click the `hello.html` file and open the console and you will have a message saying `Hello World`. That's all there is to running a program with your local environment.

## Virtual Environment



Virtual Environment with VirtualBox

The main benefits of having a virtual environment to develop your application outside of having it installed in your main machine is that it gives you the power to experiment and try different experiments without necessarily having an impact on your actual machine. Not only can you create multiple instances of the same environment you can quickly shut off machines and bring more them online, as your application requires them. This is great if one of your interests is scaling your application to multiple machines in a Virtual Environment, making it one of the first steps to moving on to a platform like AWS (Amazon Web Services).

Since the release of VirtualBox in 2007, developing applications has never been the same. If you're anything like me, you probably don't always want to be programming in the same language nor programming on the same machine. You want to see what happens when you run your application on multiple platforms. Virtual and cloud environments allow you to do that for little to no cost. Cloud

environments can be created for as low as \$5.00 USD a month, or virtual environments can be hosted on your machine as much as your hardware will allow. In this section we will explore virtual environments and find out why and when you should use them.

The current market place has a wide variety of virtualization options that allow you to use your current hardware to virtualize a machine. In this section I will only be covering two of the most common tools used by JavaScript developers.

1. [VirtualBox](#)
2. [Vagrant](#)

## **Virtual Environment with VirtualBox**

Why use a virtual environment? I'm not going to go into the reasons on why VE is important or why you should be using it. Instead, I will show you how it relates to software development and why this is a valid option when choosing to go with a host or local setup.

Here is a quick list of some of the advantages.

- Multiple instances of the same operating system in different states.
- Quickly backup and restore instances, great for working with teams.
- Only run the virtual machine when it's needed instead of having daemons running all the time.
- Keeps your local machine clean.
- Adds another level of security in case your machine is stolen.
- Allows you to create test cases to actual domain names with host files.
- Allows for a sandbox where you can have machine to machine interactions.
- Cross platform development with VDI's.

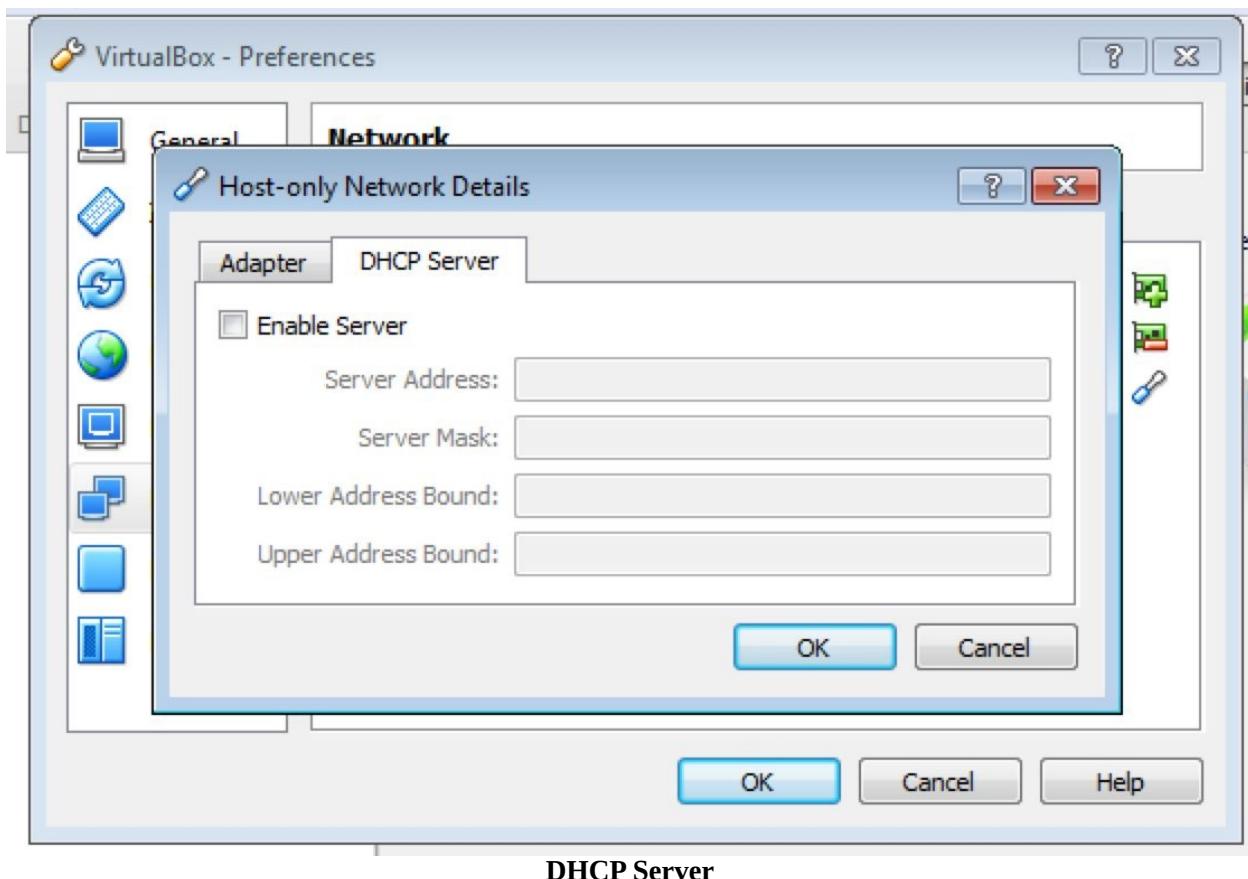
VirtualBox is defined as follows: >VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2.

The main take away from VE is to understand that virtual environments are just a tool to keep dependencies of different projects contained in a virtual machine.

## Setting up VirtualBox

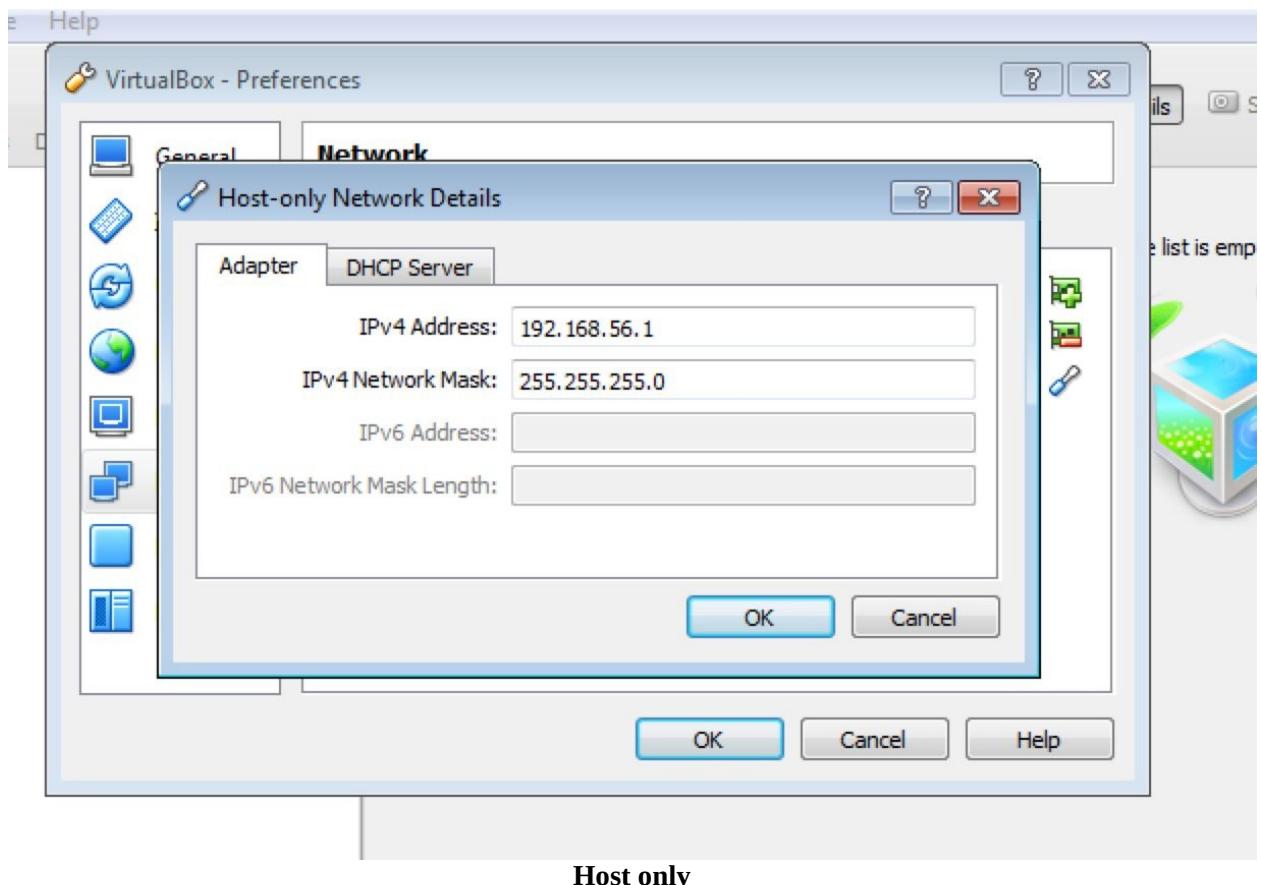
### Windows

Start by going over to the main site for [VirtualBox Downloads](#) and selecting the following option VirtualBox 5.0 for Windows hosts. This starts a download with a file name VirtualBox-5.\*.exe. Double-click the file and follow the setup wizard. This will require your machine to restart. After the machine restarts, configure VirtualBox with the following settings by going to file and clicking on Preferences. Next, click on network, click on Host-only Networks, and click on edit and disable Enable Server.



DHCP Server

The adapter host-only tab should only be modified if you have subnetworks to other machines. Otherwise, the default settings are fine.



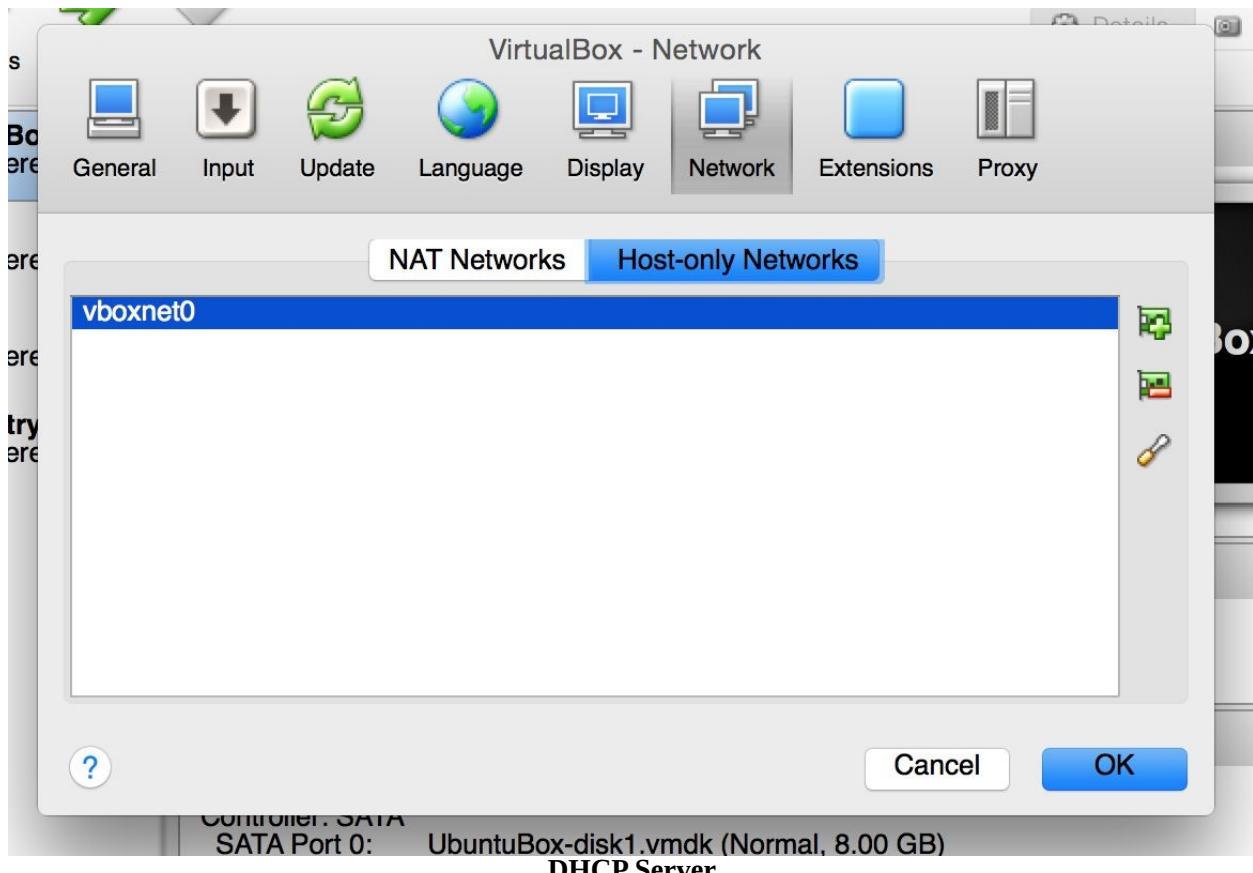
### Host only

Now VirtualBox is ready to be setup with Ubuntu.

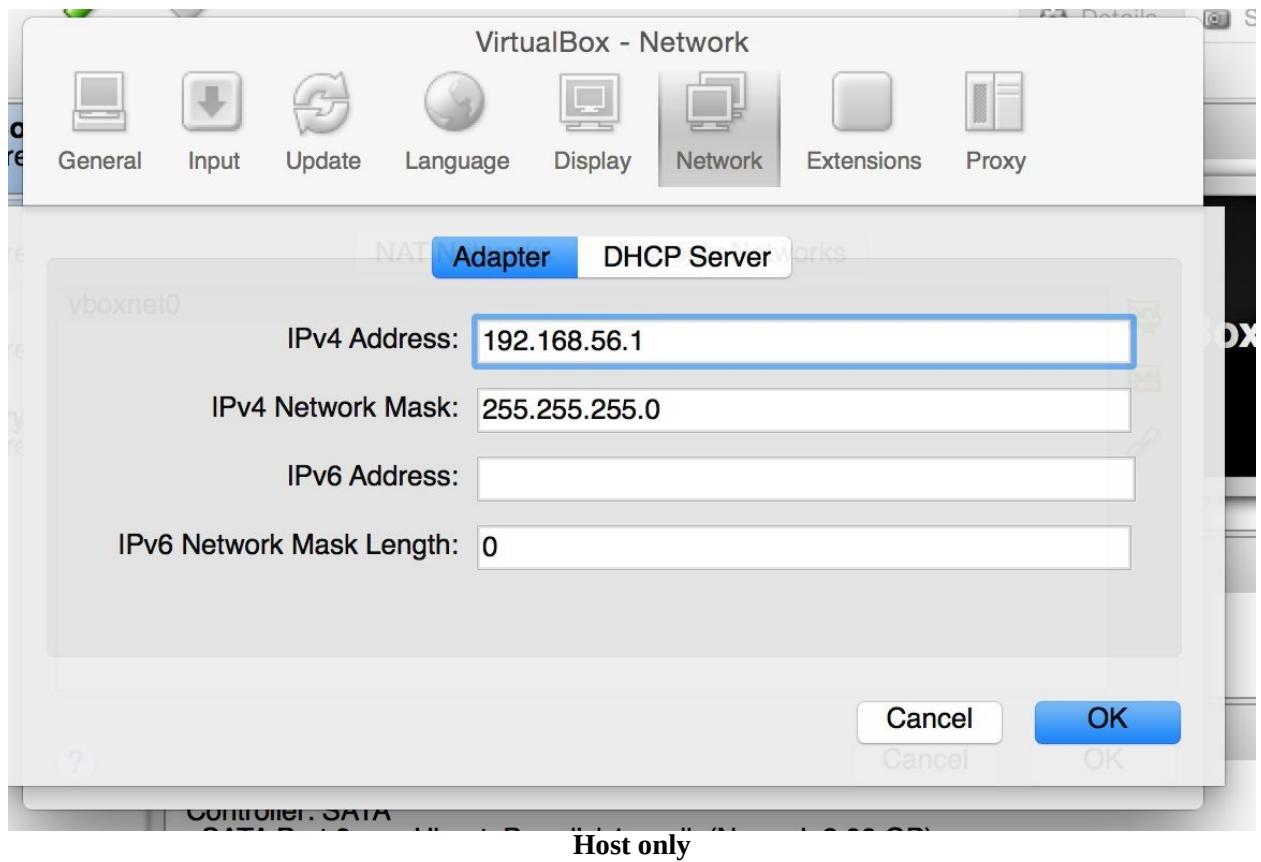
### OSX

Start by going over to the main site for [VirtualBox Downloads](#) and selecting the following option **VirtualBox 5.0 for OS X hosts** this starts a download with a file name **VirtualBox-5.\*.dmg** double click the file and follow the setup wizard. This will require your machine to restart.

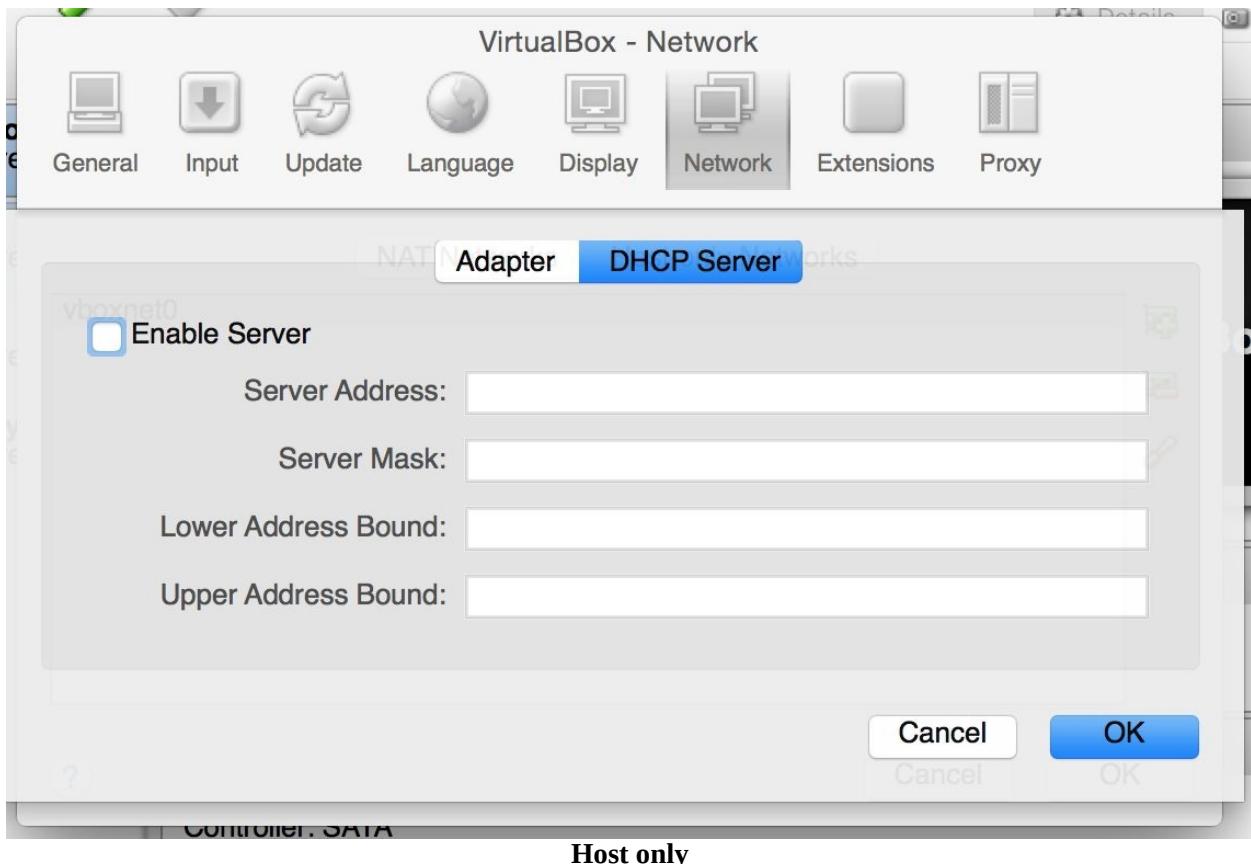
After the reboot open the program and go to **virtualBox** and click on **Prefrences**. Next, click on **network**, click on **Host-only Networks**, and click on **add host-only network(Ins)**



Now, click on edit host-only network(Ins). The adapter settings should only be modified if you have subnetworks to other machines. Otherwise, the default settings are fine.



On the DHCP Server tab make sure that this option is **disabled**.

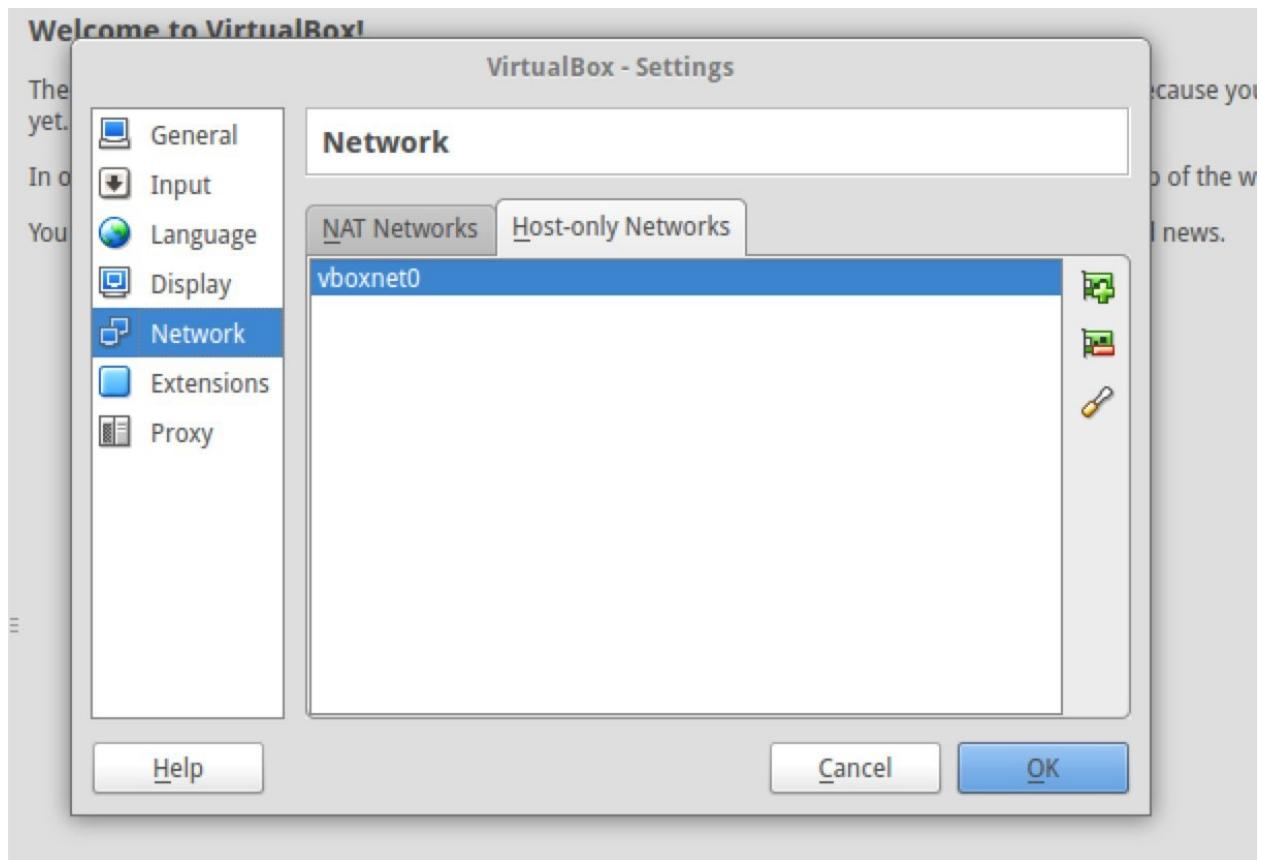


Now VirtualBox is ready to be set up with Ubuntu.

### Linux

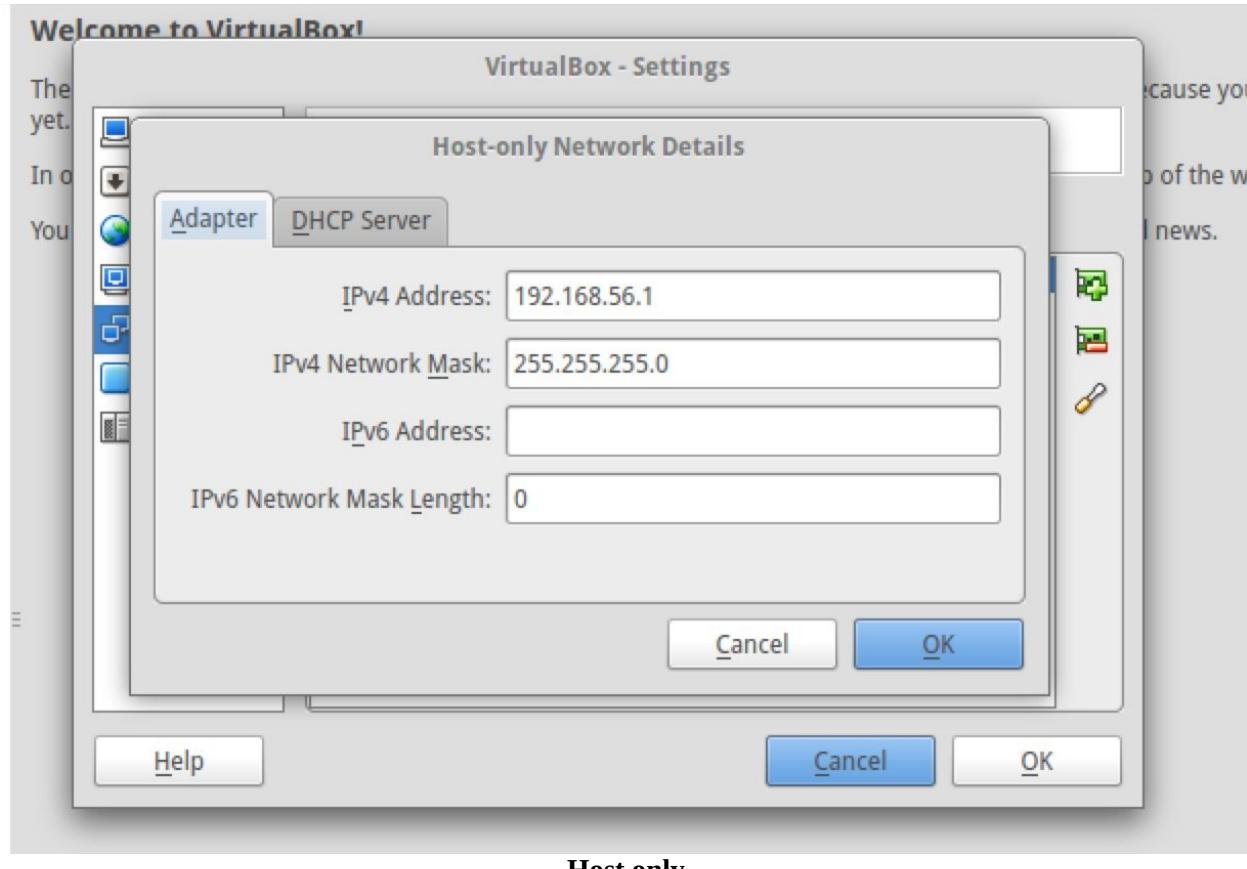
Start by going over to the main site for [VirtualBox Downloads](#) and selecting the following option `VirtualBox 5.0 for Linux hosts`. This link will take you to a selection for what type of Linux version you are using. In my case, I'm using an Ubuntu-based system. This starts a download with a file name `VirtualBox-5.*.deb`. Double-click it to open the software center. Select `install` and wait for the install to finish. This will require your machine to restart.

After the reboot, open the program and go to `file` and click on `Prefrences`. Next, click the `Host-only Networks` and click on `add host-only network(Ins)`.



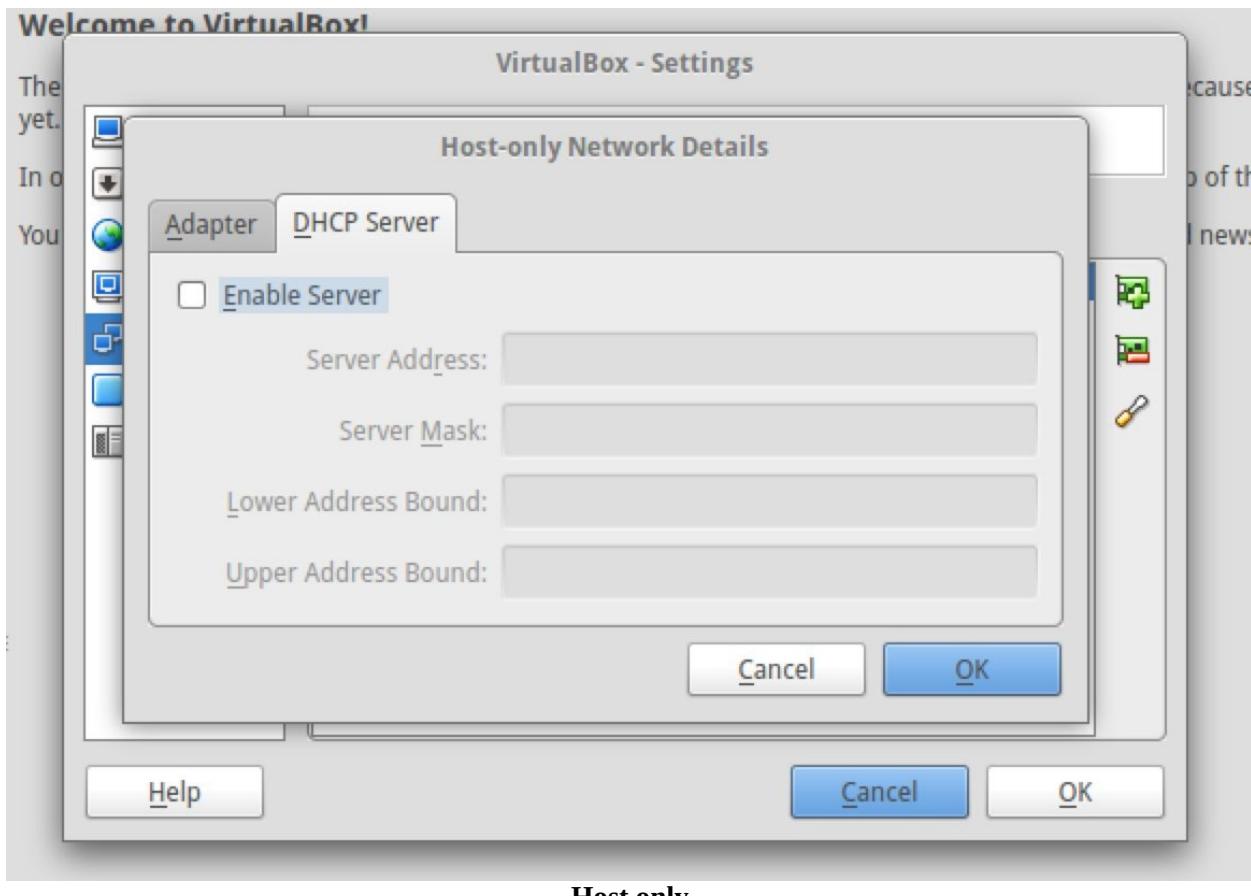
### DHCP Server

Now, click on edit host-only network(Ins). The adapter settings should only be modified if you have subnetworks to other machines. Otherwise, the default settings are fine.



### Host only

On the DHCP Server tab make sure that this option is **disabled**.



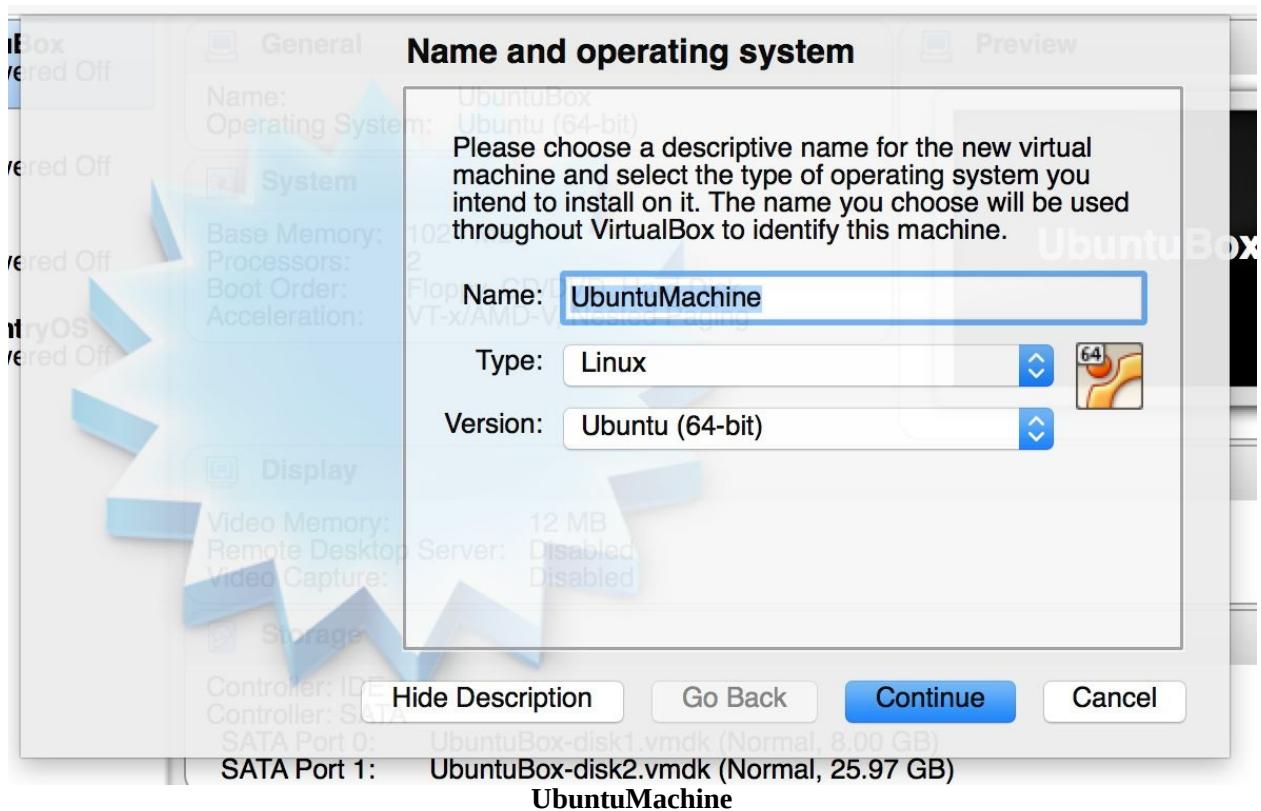
### Host only

Now VirtualBox is ready to be set up with Ubuntu.

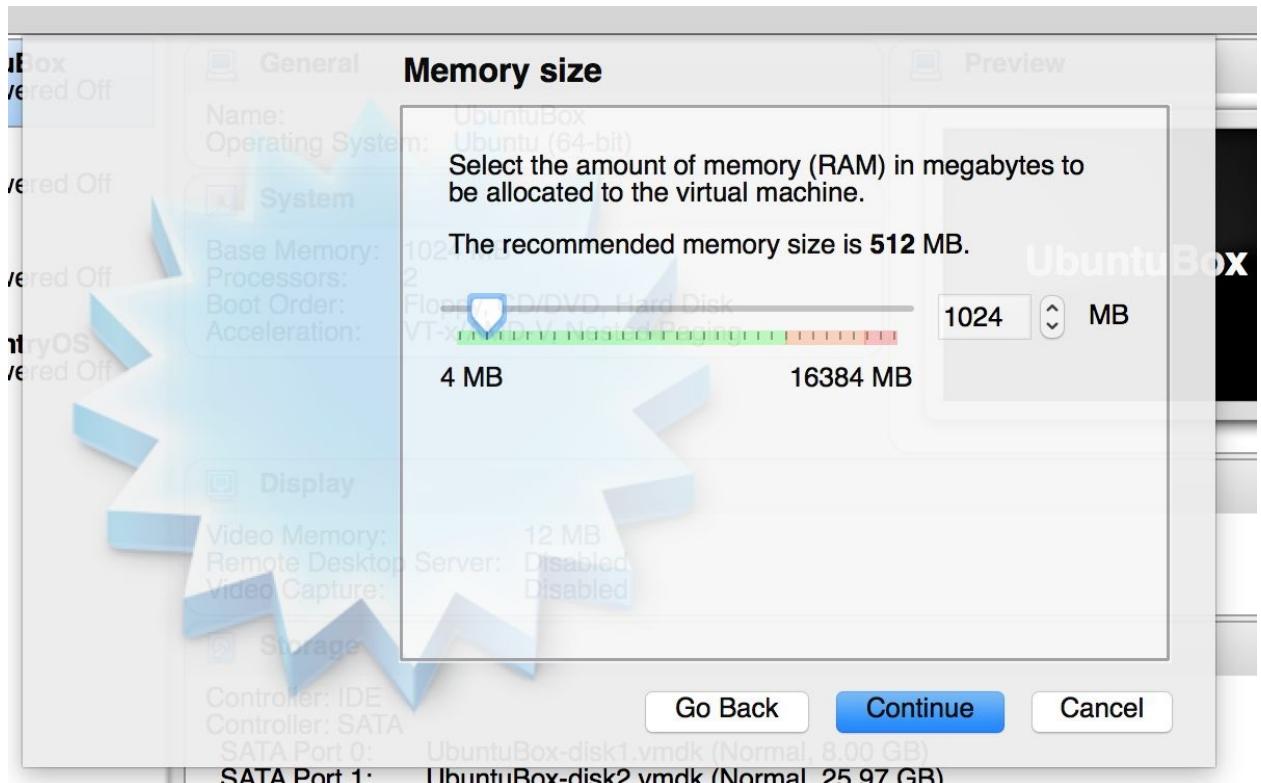
## Installing Ubuntu with VirtualBox

Before getting started with VirtualBox, you are going to have to grab a fresh copy of Ubuntu from their main [site](#). Get a version that has LTS(Long term support). After the download is complete, you should have a file named `ubuntu-14.04.2-server-amd64.iso`.

Open VirtualBox and select the New option from the top navigation. Give the machine a name such as `UbuntuMachine`.

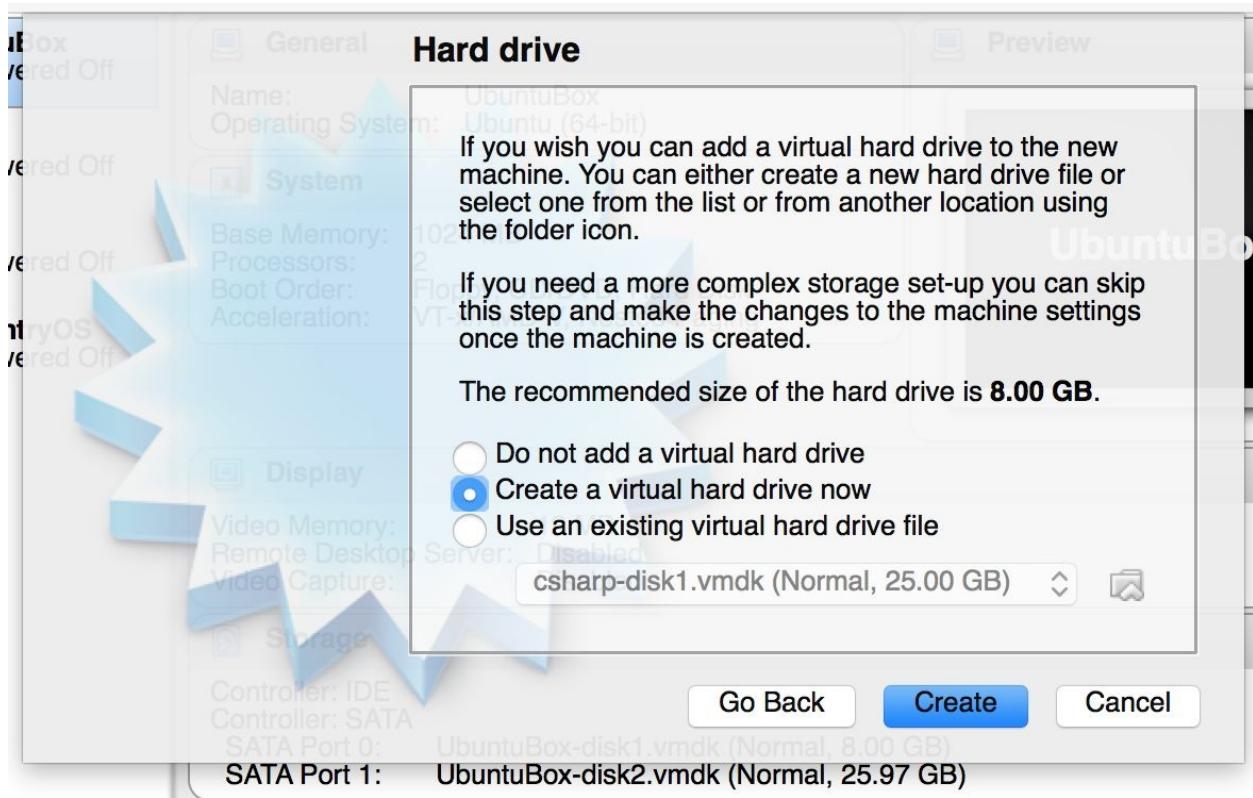


Click on Continue, Memory size I tend to use **1024MB**



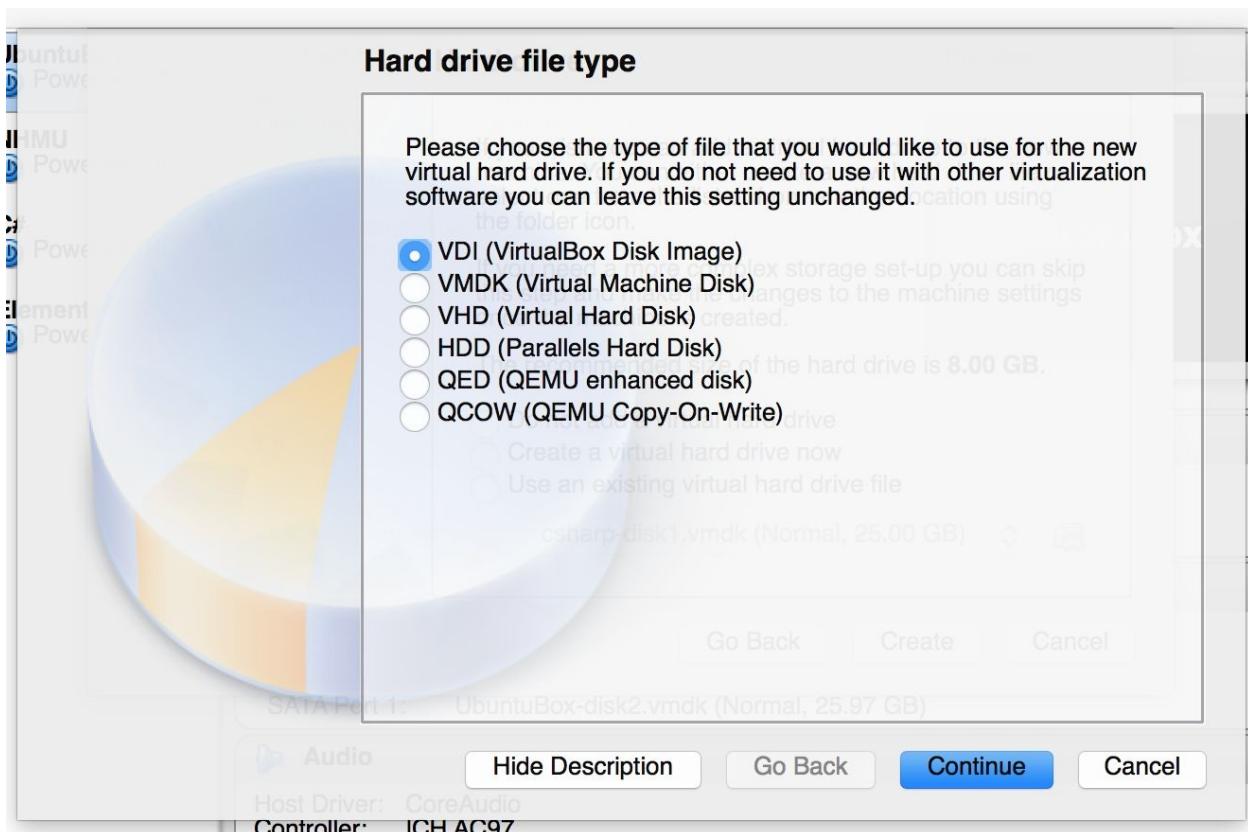
### UbuntuMachine

Click on Continue, Select Create a virtual hard drive now

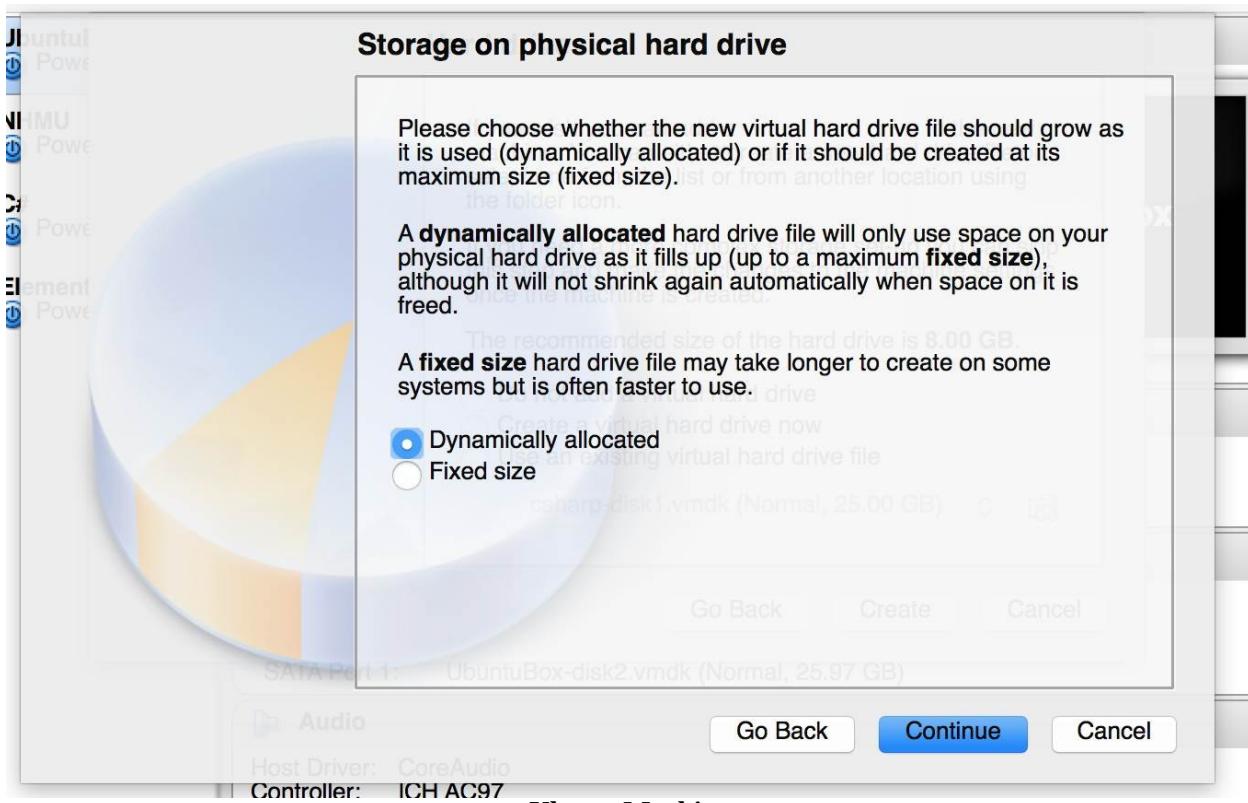


UbuntuMachine

Click on Create, Select VDI(Virtual Disk Image)



Click on Continue, Select Dynamically Allocated

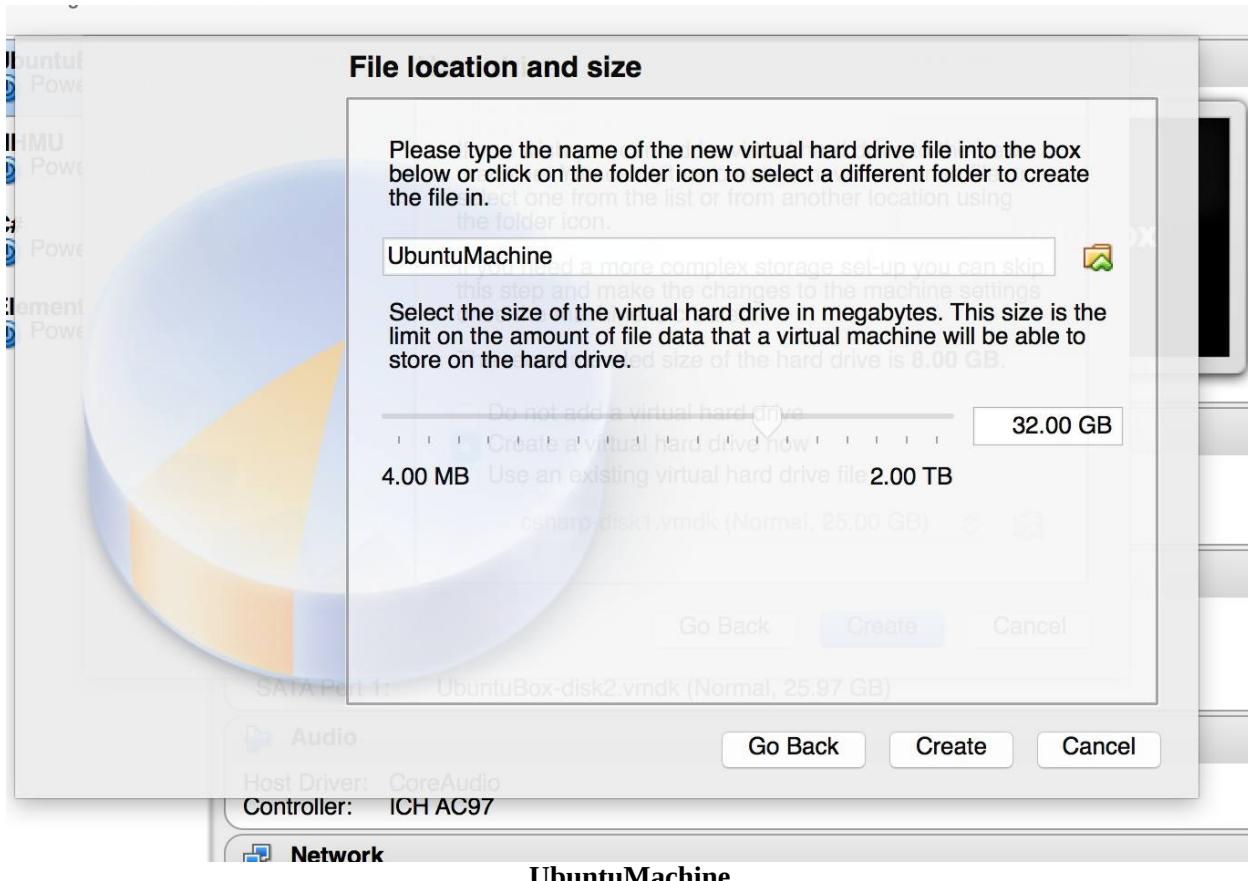


Click on Continue, Hard drive size I tend to use **32GB**



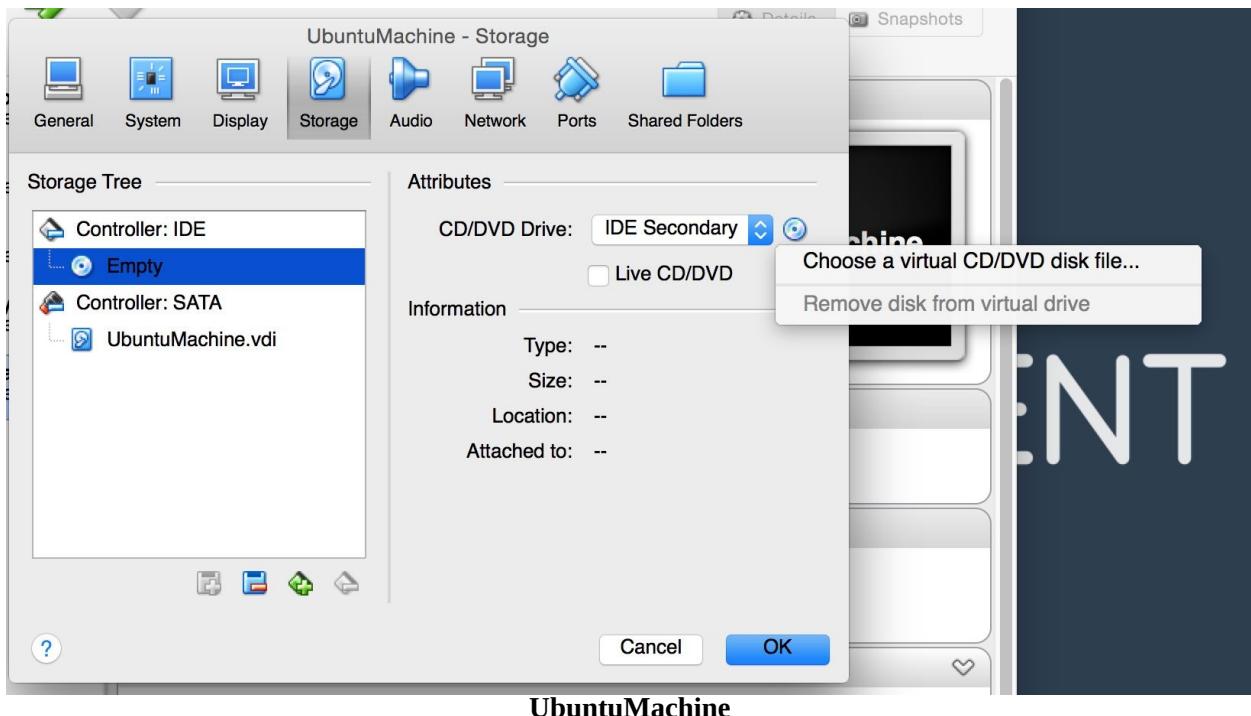
### Warning

When creating a new virtual machine, think about the resource allocation that the virtual machine will need. Is 8gb enough? what about 32gb?.



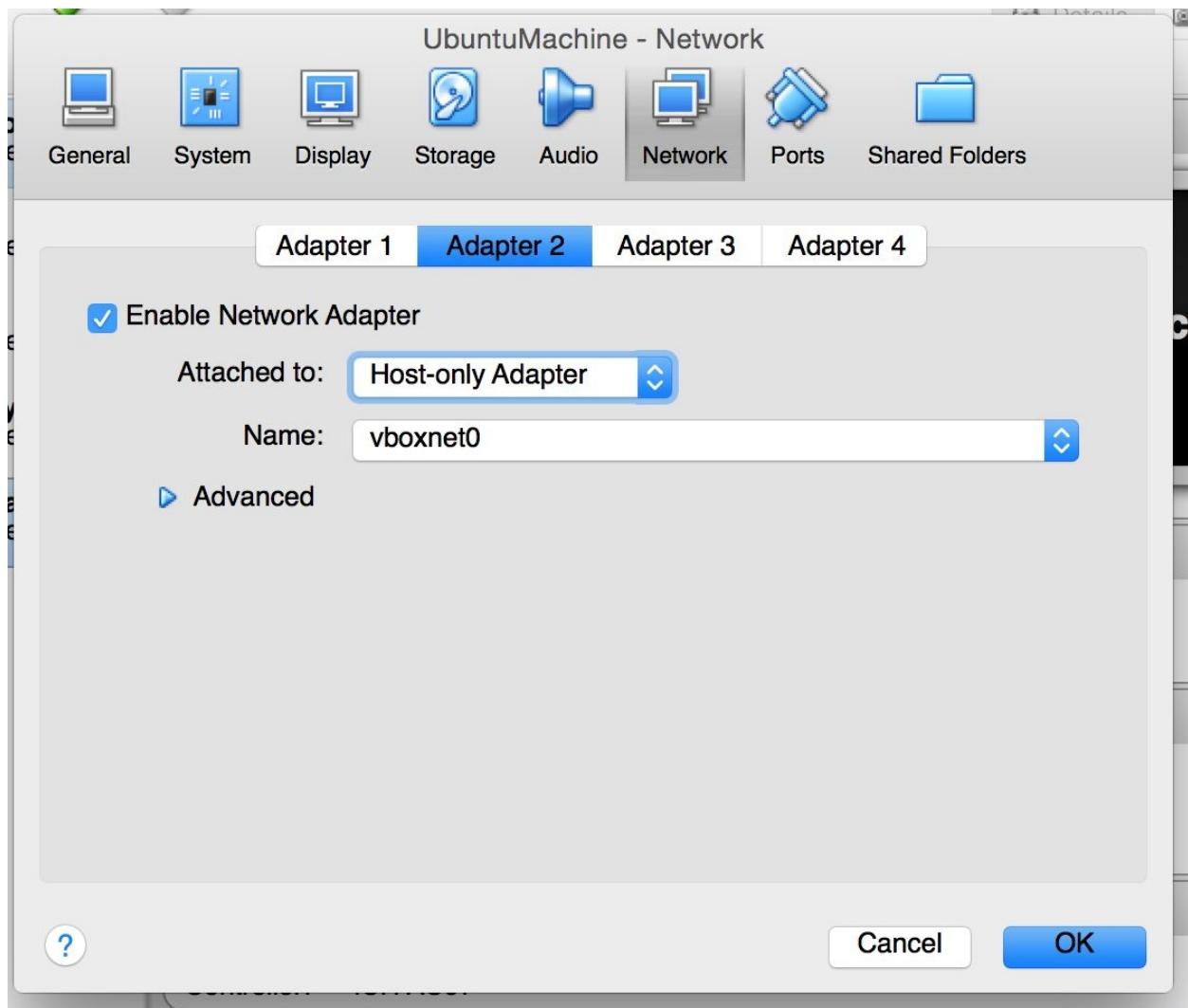
## Select **Create**

Now set up the .iso image so that the VM can see it. Select the machine and click on settings. Select Storage from the top navigation. Select the empty Controller IDE and choose Virtual CD/DVD disk file, then click on Ok.



UbuntuMachine

Before moving on to the next step, make sure that you enable the **Host-Only File**.



host file

Let's start up the machine by clicking on Start. This will start the virtual machine. Now follow the instructions to set up Ubuntu.

Language			
Amharic	Français	Македонски	Tamil
Arabic	Gaeilge	Malayalam	ଓଡ଼ିଆ
Asturianu	Galego	Marathi	Thai
Беларуская	Gujarati	Burmese	Tagalog
Български	ગુજરાતી	Nepali	Türkçe
Bengali	Hindi	Nederlands	Uyghur
Tibetan	Hrvatski	Norsk bokmål	Українська
Bosanski	Magyar	Norsk nynorsk	Tiếng Việt
Català	Bahasa Indonesia	Punjabi (Gurmukhi)	中文(简体)
Čeština	Íslenska	Polski	中文(繁體)
Dansk	Italiano	Português do Brasil	
Deutsch	日本語	Português	
Dzongkha	ཇանওঞ্চো	Română	
Ελληνικά	Қазақ	Русский	
<b>English</b>	Khmer	Sámegillii	
Esperanto	ಕನ್ನಡ	ଓ.ହାତ	
Español	한국어	Slovenčina	
Eesti	Kurdî	Slovenščina	
Euskara	Lao	Shqip	
فارسی	Lietuviškai	Српски	
Suomi	Latviski	Svenska	

F1 Help F2 Language F3 Keymap F4 Modes F5 Accessibility F6 Other Options

**Setup Linux Ubuntu**

# ubuntu®

Install Ubuntu Server  
Multiple server install with MAAS  
Check disc for defects  
Test memory  
Boot from first hard disk  
Rescue a broken system

F1 Help F2 Language F3 Keymap F4 Modes F5 Accessibility F6 Other Options

**Setup Linux Ubuntu**

[!!] Select a language

Choose the language to be used for the installation process. The selected language will also be the default language for the installed system.

Language:

- |                       |                   |
|-----------------------|-------------------|
| C                     | - No localization |
| Albanian              | - Shqip           |
| Arabic                | - العربية         |
| Asturian              | - Asturianu       |
| Basque                | - Euskara         |
| Belarusian            | - Беларуская      |
| Bosnian               | - Bosanski        |
| Bulgarian             | - Български       |
| Catalan               | - Català          |
| Chinese (Simplified)  | - 中文(简体)          |
| Chinese (Traditional) | - 中文(繁體)          |
| Croatian              | - Hrvatski        |
| Czech                 | - Čeština         |
| Danish                | - Dansk           |
| Dutch                 | - Nederlands      |
| English               | - English         |
| Esperanto             | - Esperanto       |
| Estonian              | - Eesti           |
| Finnish               | - Suomi           |
| French                | - Français        |
| Galician              | - Galego          |
| German                | - Deutsch         |
| Greek                 | - Ελληνικά        |

<Go Back>

Setup Linux Ubuntu

[!] Select your location

The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.

This is a shortlist of locations based on the language you selected. Choose "other" if your location is not listed.

Country, territory or area:

- Antigua and Barbuda
- Australia
- Botswana
- Canada
- Hong Kong
- India
- Ireland
- New Zealand
- Nigeria
- Philippines
- Singapore
- South Africa
- United Kingdom
- United States**
- Zambia
- Zimbabwe
- other

[<Go Back>](#)

**Setup Linux Ubuntu**

[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

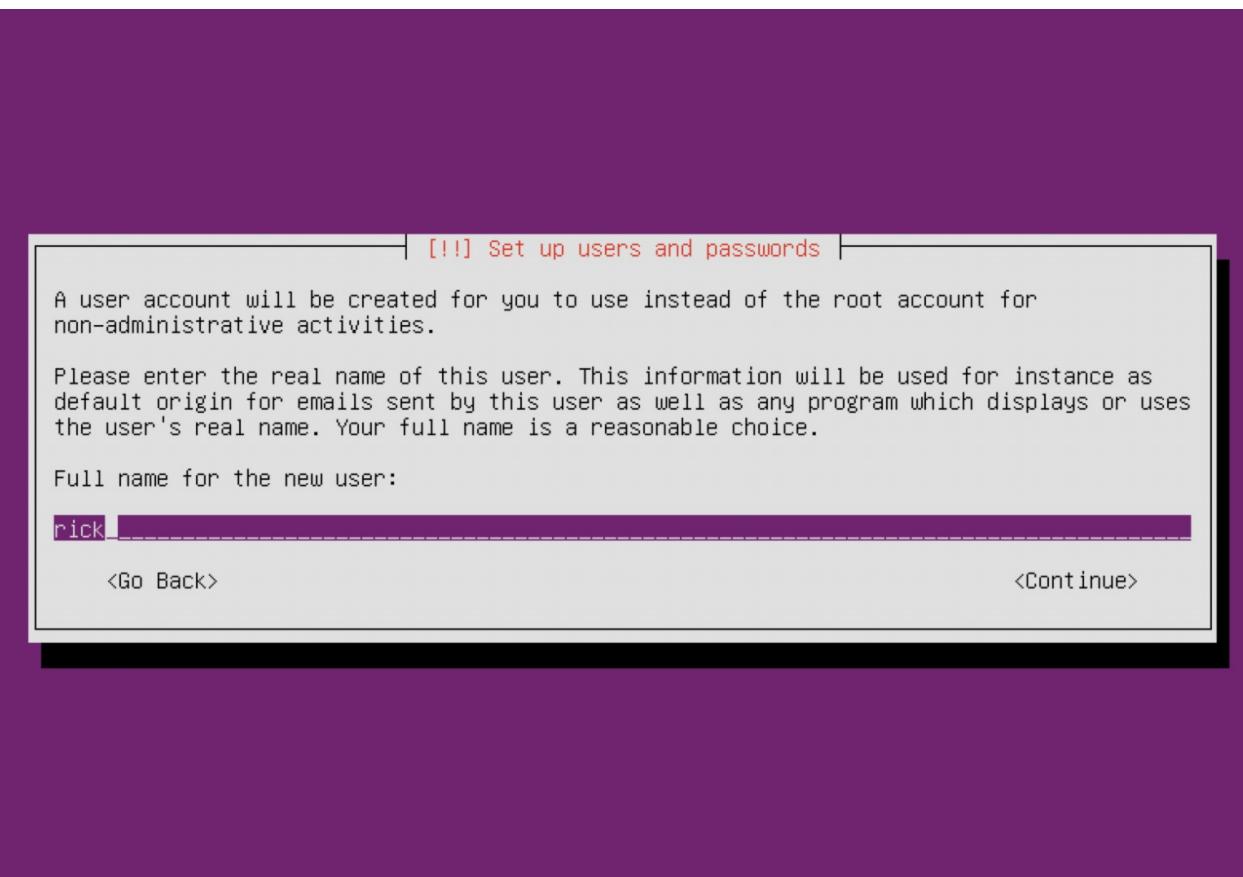
ubuntu\_

<Go Back>

<Continue>

Tab> moves; <Space> selects; <Enter> activates buttons

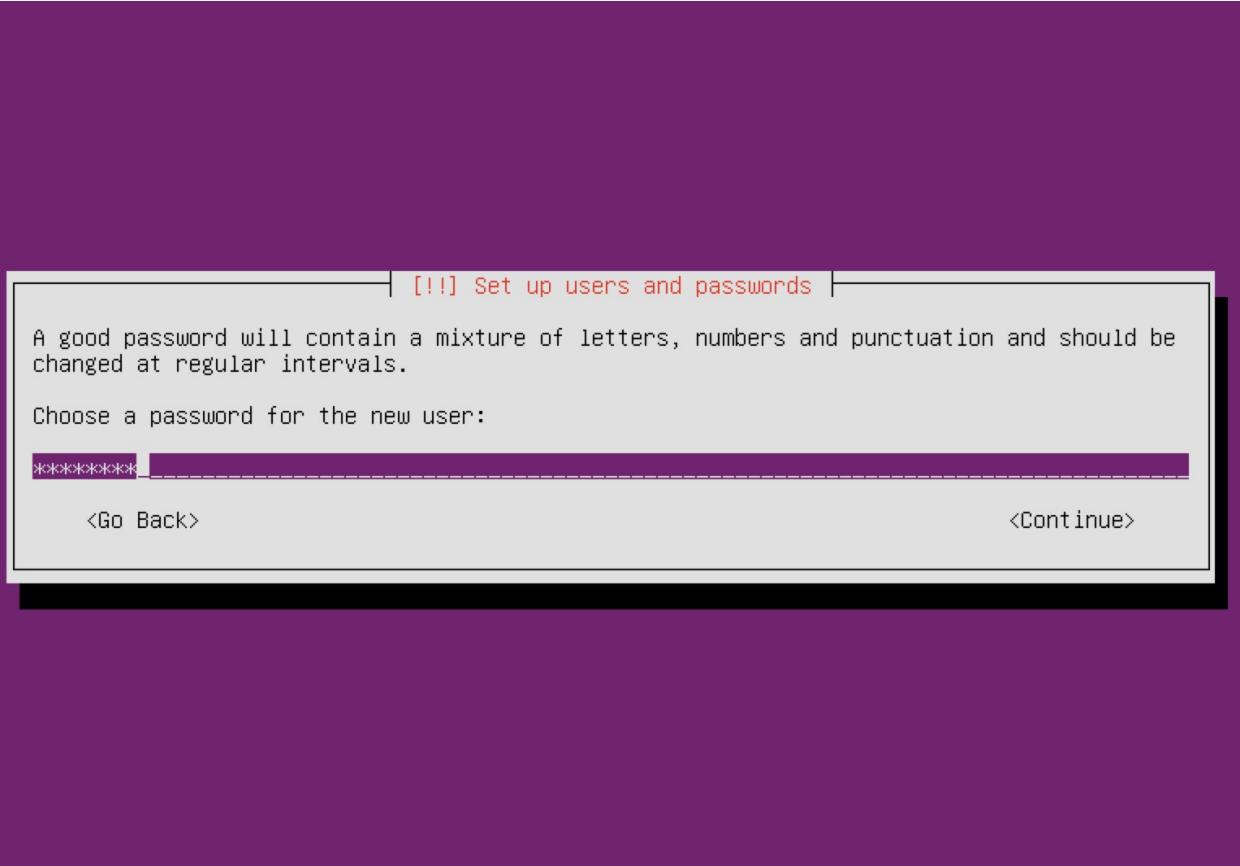
**Setup Linux Ubuntu**



**Setup Linux Ubuntu**



**Setup Linux Ubuntu**



**Setup Linux Ubuntu**

### [!] Set up users and passwords

You may configure your home directory for encryption, such that any files stored there remain private even if your computer is stolen.

The system will seamlessly mount your encrypted home directory each time you login and automatically unmount when you log out of all active sessions.

Encrypt your home directory?

<Go Back>

<Yes>

<No>

## Setup Linux Ubuntu

### [!!] Partition disks

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

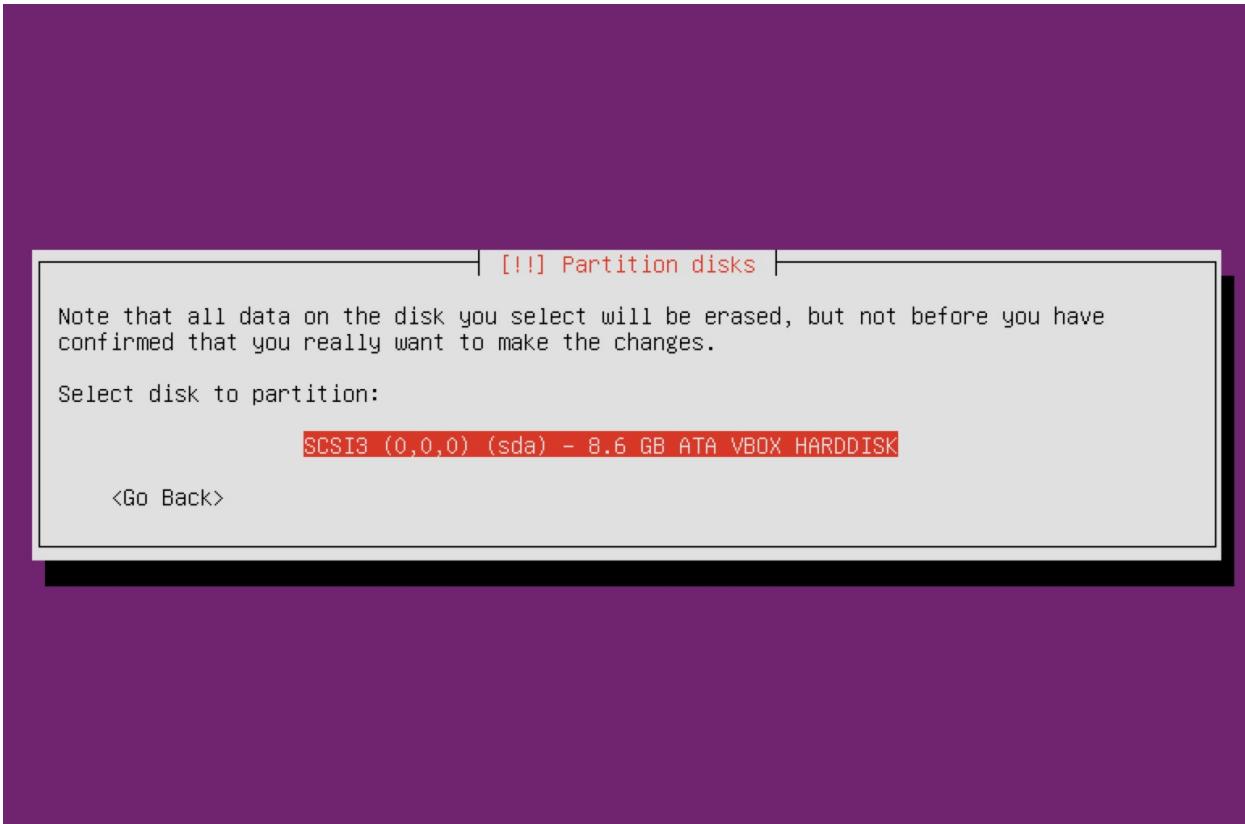
If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

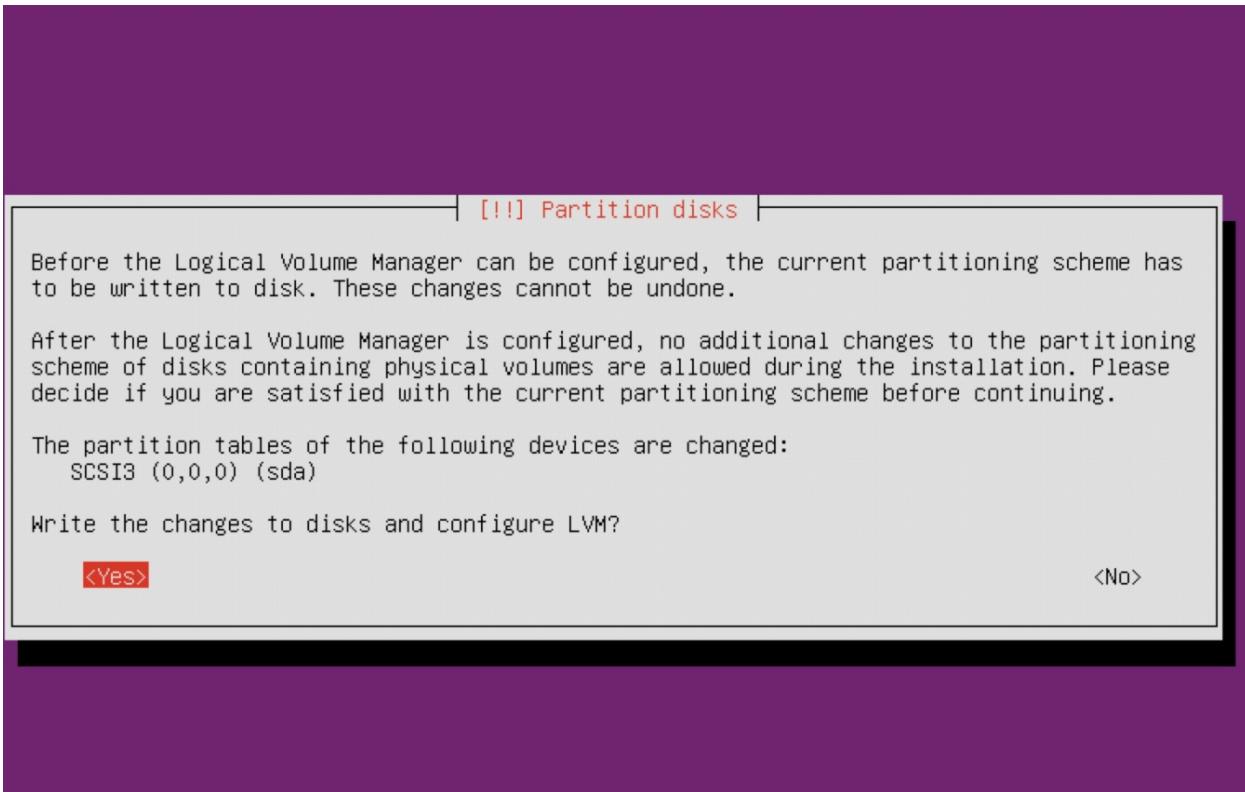
- Guided - use entire disk
- Guided - use entire disk and set up LVM
- Guided - use entire disk and set up encrypted LVM
- Manual

<Go Back>

## Setup Linux Ubuntu



### Setup Linux Ubuntu



### Setup Linux Ubuntu

[!] Partition disks

You may use the whole volume group for guided partitioning, or part of it. If you use only part of it, or if you add more disks later, then you will be able to grow logical volumes later using the LVM tools, so using a smaller part of the volume group at installation time may offer more flexibility.

The minimum size of the selected partitioning recipe is 2.0 GB (or 23%); please note that the packages you choose to install may require more space than this. The maximum available size is 8.3 GB.

Hint: "max" can be used as a shortcut to specify the maximum size, or enter a percentage (e.g. "20%") to use that percentage of the maximum size.

Amount of volume group to use for guided partitioning:

8.3 GB

<Go Back>

<Continue>

Setup Linux Ubuntu

[!!] Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

The partition tables of the following devices are changed:

LVM VG ubuntu-vg, LV root  
LVM VG ubuntu-vg, LV swap\_1  
SCSI3 (0,0,0) (sda)

The following partitions are going to be formatted:

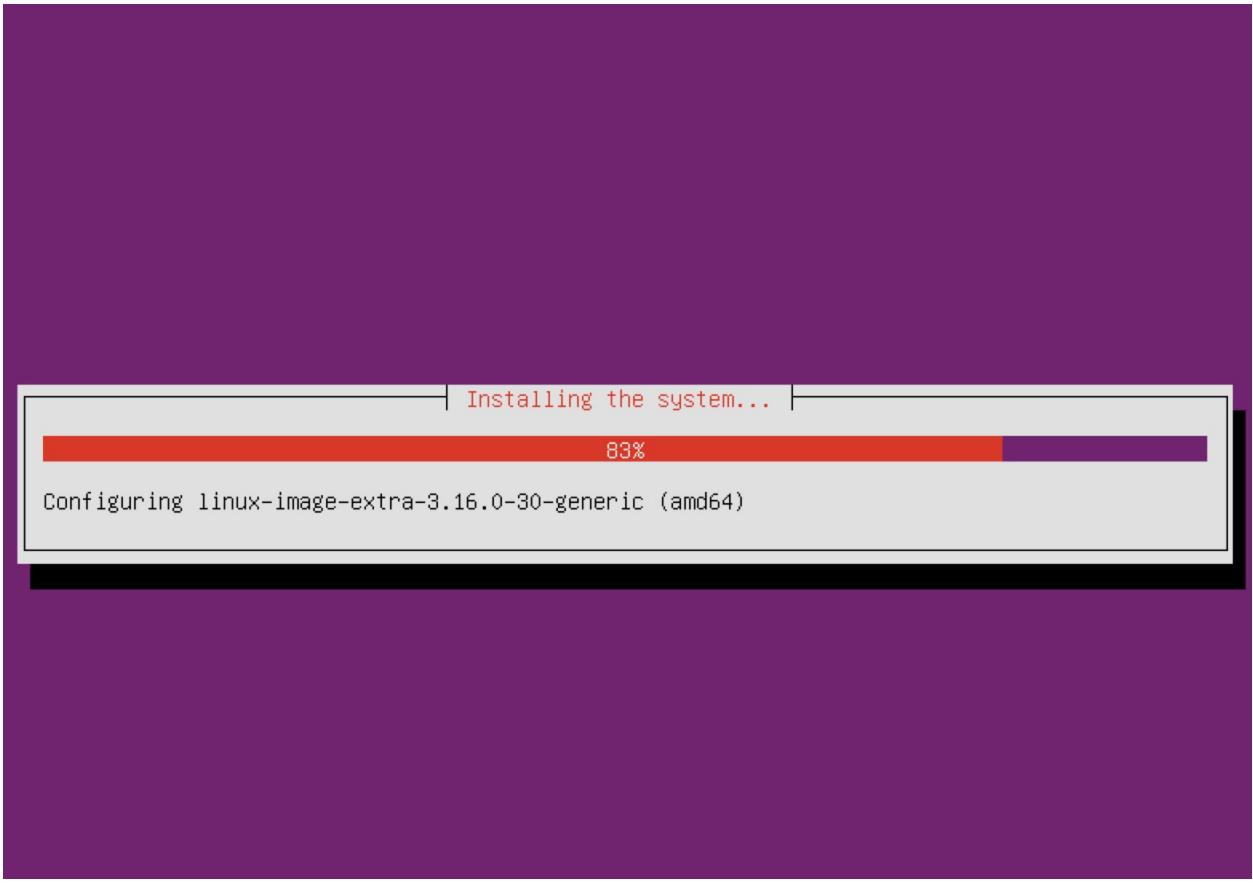
LVM VG ubuntu-vg, LV root as ext4  
LVM VG ubuntu-vg, LV swap\_1 as swap  
partition #1 of SCSI3 (0,0,0) (sda) as ext2

Write the changes to disks?

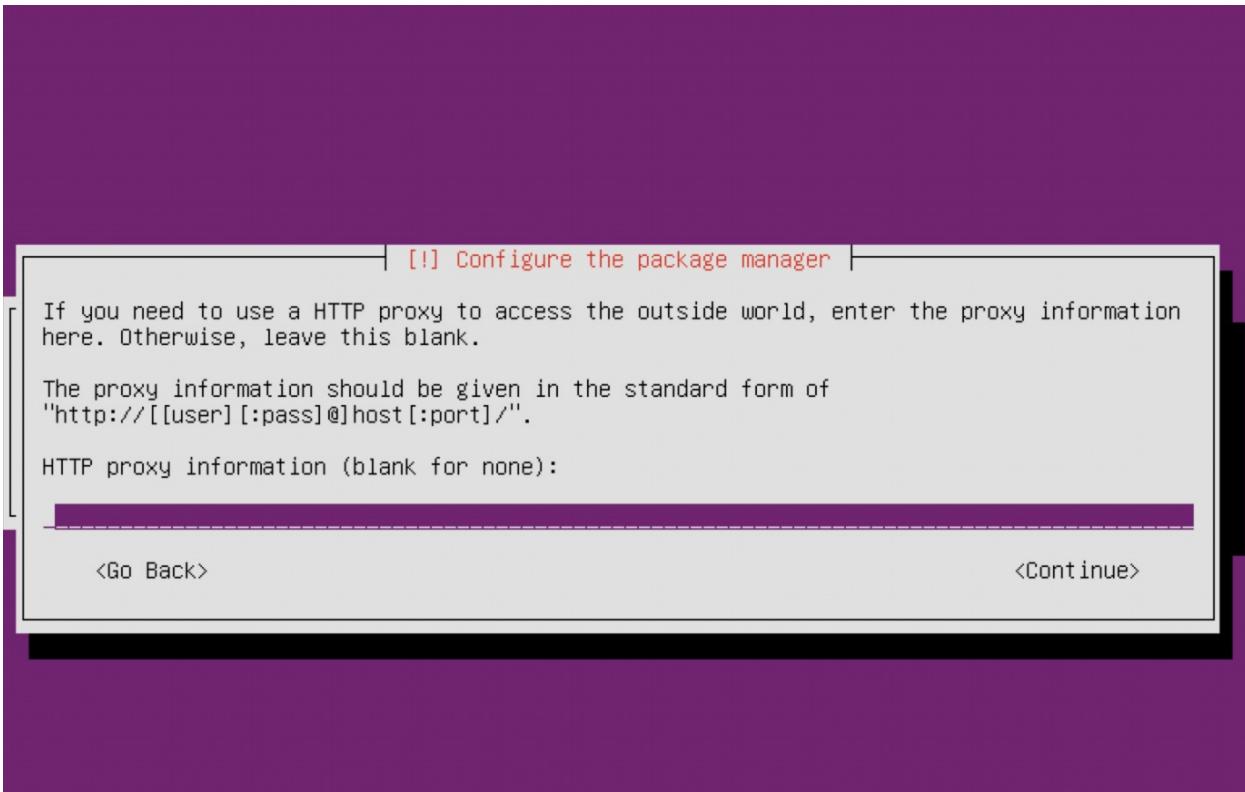
<Yes>

<No>

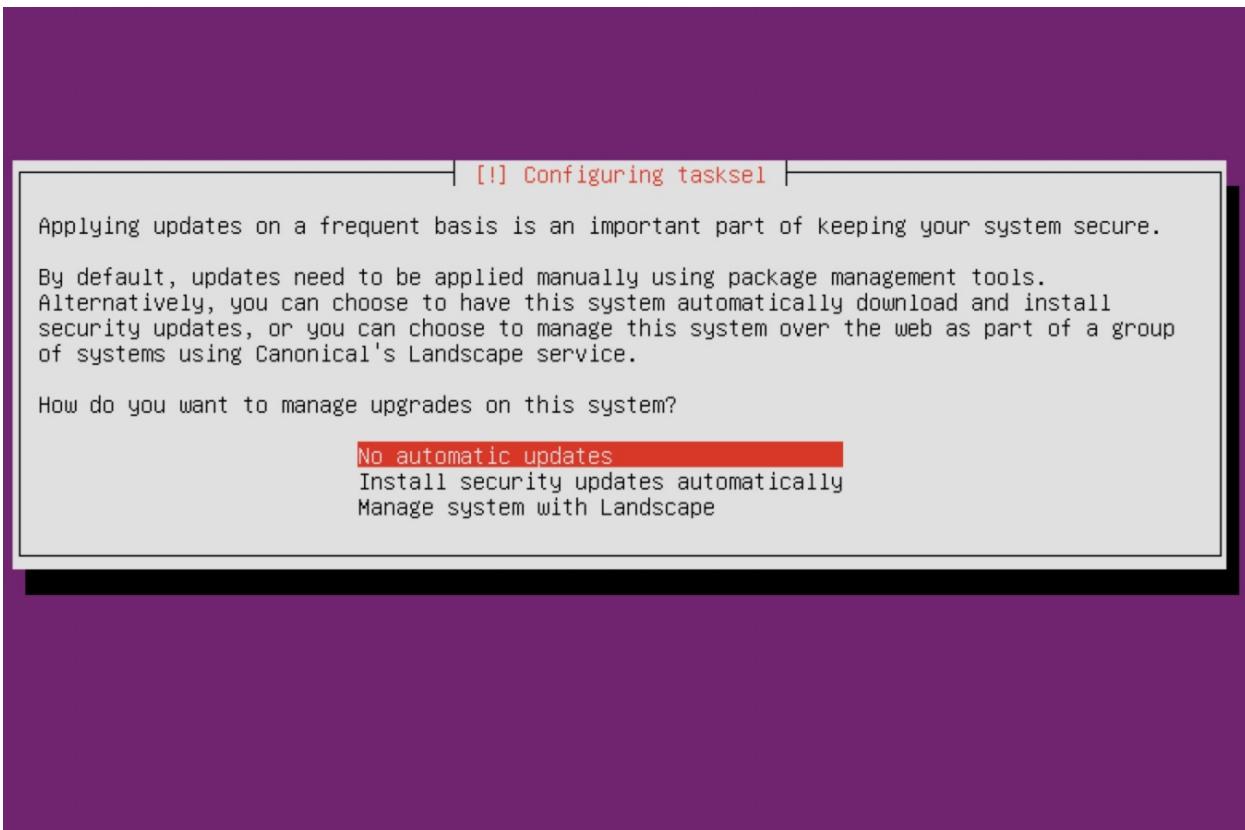
**Setup Linux Ubuntu**



**Setup Linux Ubuntu**



## Setup Linux Ubuntu



## Setup Linux Ubuntu

### [!] Software selection

At the moment, only the core of the system is installed. To tune the system to your needs, you can choose to install one or more of the following predefined collections of software.

Choose software to install:

- OpenSSH server
- DNS server
- LAMP server
- Mail server
- PostgreSQL database
- Print server
- Samba file server
- Tomcat Java server
- Virtual Machine host
- Manual package selection

[<Continue>](#)

## Setup Linux Ubuntu

### [!] Install the GRUB boot loader on a hard disk

It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to the master boot record of your first hard drive.

Warning: If the installer failed to detect another operating system that is present on your computer, modifying the master boot record will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to the master boot record?

[<Go Back>](#)

[<Yes>](#)    [<No>](#)

## Setup Linux Ubuntu



## Setup Linux Ubuntu

### interfaces

```
1 sudo vim /etc/network/interfaces
```

---

```
ubuntu login: rick
Password:
no talloc stackframe at ../source3/param/loadparm.c:4864, leaking memory
Added user rick.

Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)

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

 System information as of Tue Jul 21 18:54:07 CDT 2015

 System load: 0.0           Memory usage: 5%   Processes:      83
 Usage of /: 17.1% of 6.50GB Swap usage:  0%   Users logged in: 0

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

123 packages can be updated.
75 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

rick@ubuntu:~$
```

## Setup Linux Ubuntu

### interfaces

```
1 auto eth1
2 iface eth1 inet static
3 address 192.168.56.101
4 netmask 255.255.255.0
```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.56.101
netmask 255.255.255.0
~
```

## Setup Linux Ubuntu

reboot

```
1 sudo reboot
```

At this point you should have access to the machine via the terminal.

ssh

```
1 ssh username@192.168.56.101
```

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.101' (RSA) to the list of known hosts.
rick@192.168.56.101's password:
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.16.0-30-generic x86_64)

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

 System information as of Tue Jul 21 19:05:09 CDT 2015

 System load:  0.0          Processes:      90
 Usage of /:   17.2% of 6.50GB  Users logged in:  0
 Memory usage: 5%           IP address for eth1: 192.168.56.101
 Swap usage:   0%

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

123 packages can be updated.
75 updates are security updates.

Last login: Tue Jul 21 19:04:26 2015
rick@ubuntu:~$
```

### Setup Linux Ubuntu

The next part is to configure samba to be able to share the `www` root folder in order to use it for development.

#### smb.conf

```
1 sudo vim /etc/samba/smb.conf
```

#### smb.conf

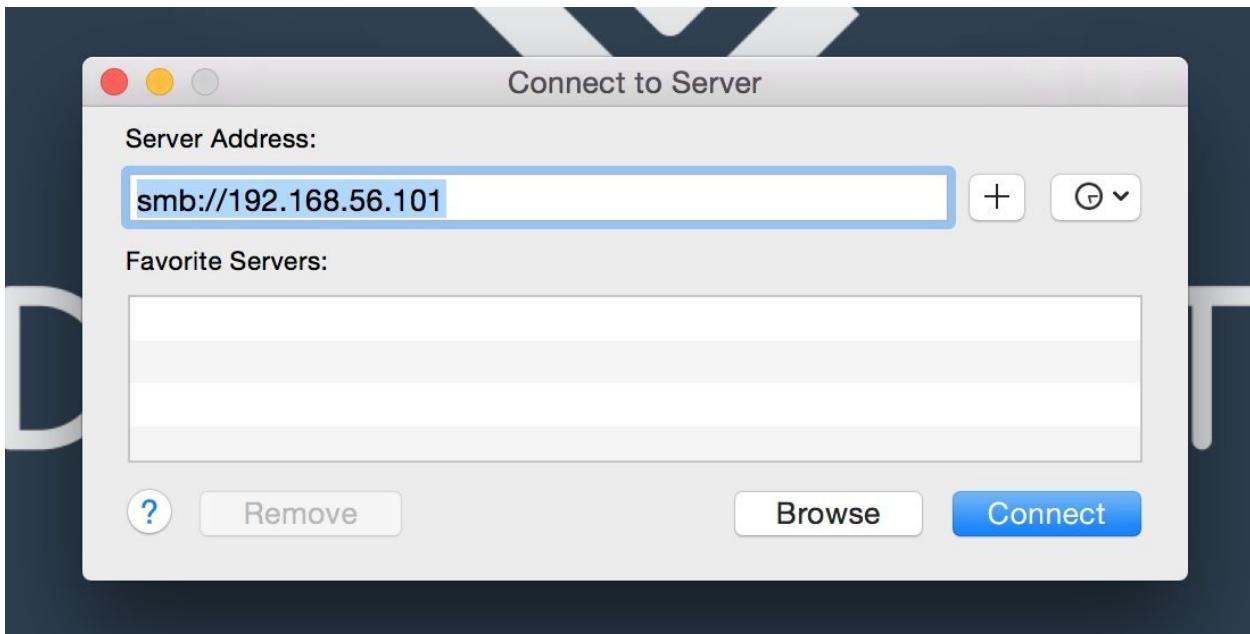
```
1 [www]
2 path = /var/www
3 public = yes
4 writable = yes
5 available = yes
6 valid users = <USERNAME>
7 create mask = 0775
8 directory mask = 0775
9 force user = <USERNAME>
10 force directory = www-data
11 follow symlinks = yes
12 wide-links = yes
```

```
1 sudo mkdir /var/www
2 sudo chown <USERNAME>:<USERGROUP> /var/www
```

```
3 sudo service smbd restart
```

Now your local machine needs to know about the shared directory.

**OSX:** On a mac you can mount a samba share by using the **finder** and clicking on **Go** then **Connect to Server** then enter your credentials which will mount the share.



Setup OSX Samba Mount



Enter your name and password for the server  
“192.168.56.101”.

Connect as:  Guest

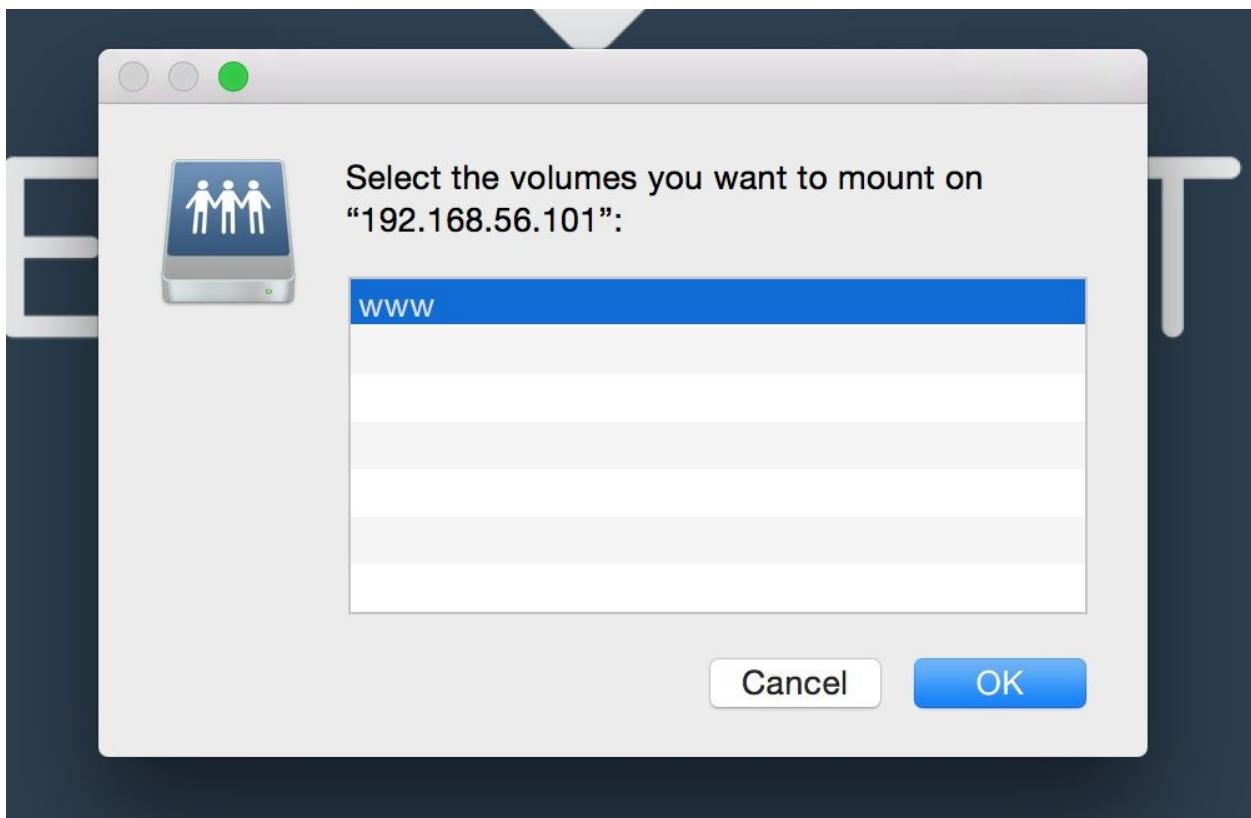
Registered User

Name:

Password:

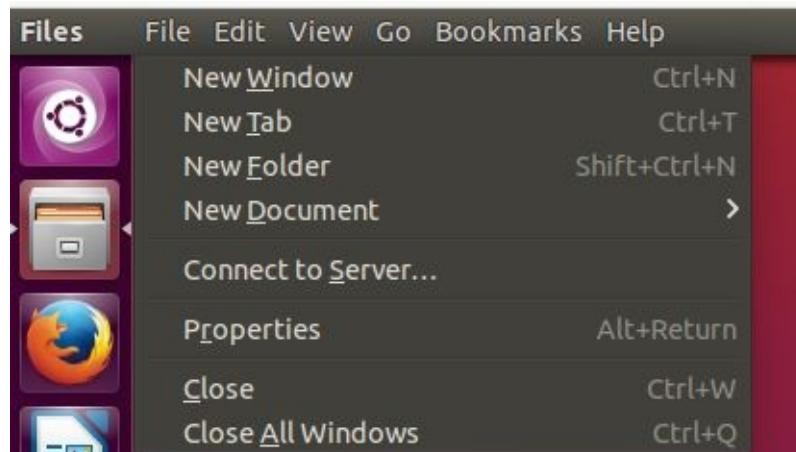
Remember this password in my keychain

**Setup OSX Samba Mount**

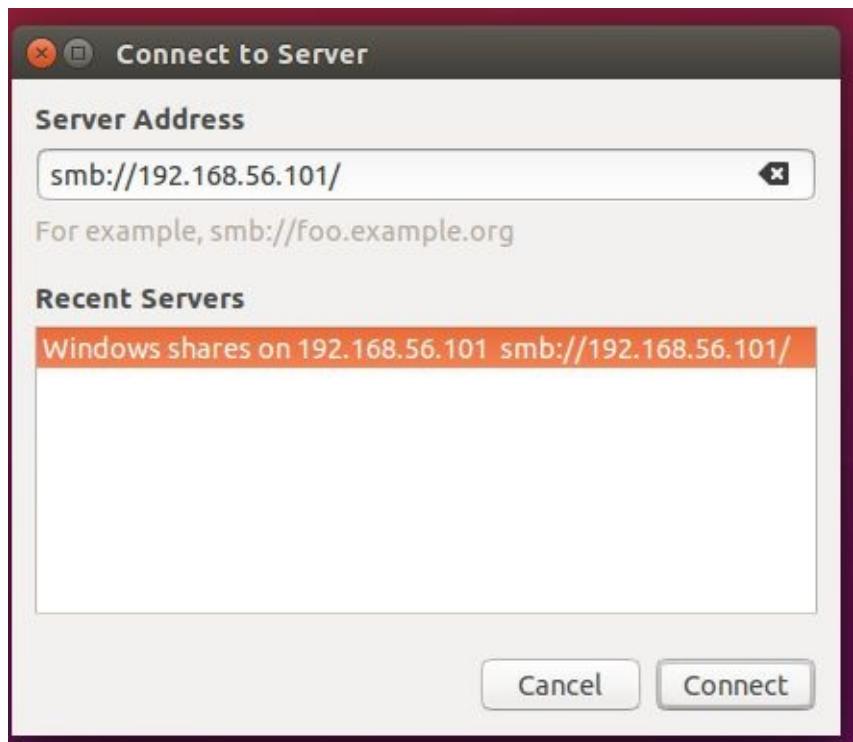


**Setup OSX Samba Mount**

**Linux:** In Ubuntu you can mount a samba share by using the **Files** application and clicking on **File** menu **Connect to Server** and enter your credentials and the share will be mounted.



**Setup Linux Ubuntu**

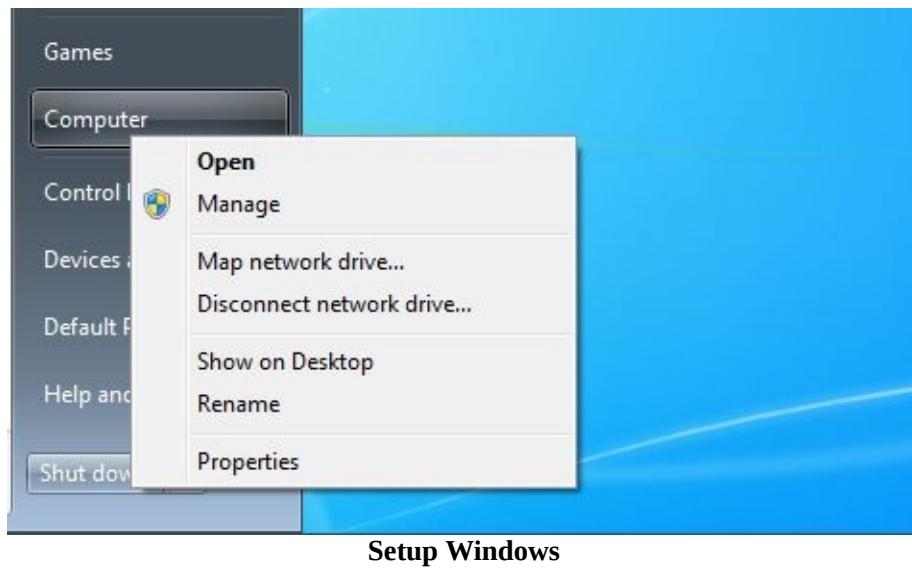


Setup Linux Ubuntu

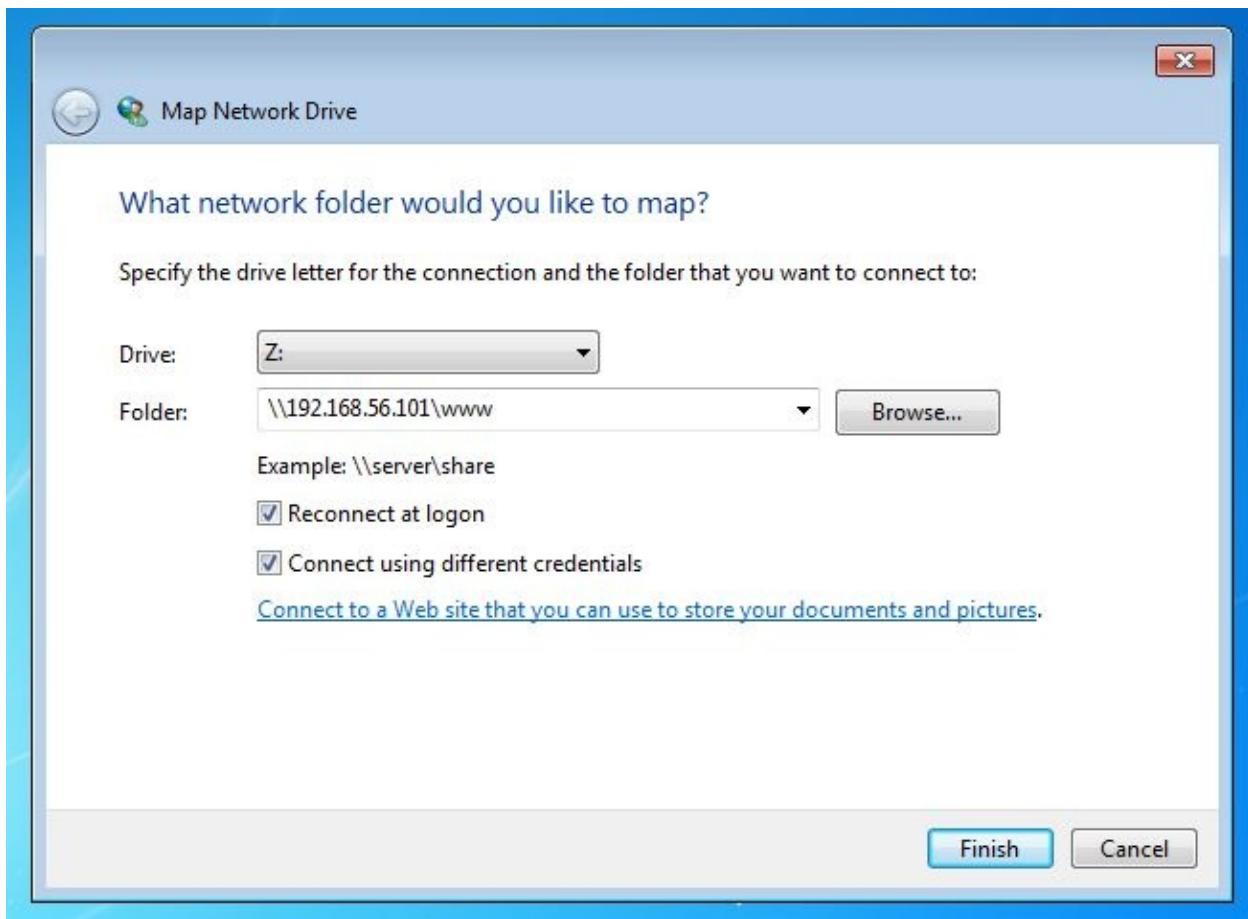


Setup Linux Ubuntu

**Windows:** In a Windows 7 you can mount a samba share by using the Start Menu and clicking on Map network drive... and entering the location of the virtual machine along with the credentials which will mount the share.



Setup Windows



Setup Windows



### Setup Windows

Now that the virtual machine is ready to be used you can start installing any software, using the command line. If you run into any issues setting up VirtualBox, you can find the complete documentation for VirtualBox using the online [manual](#).

## Virtual Environment with Vagrant

Vagrant goes a step further than VirtualBox, by automating all of the steps that you must take in order to create a virtual VM. You no longer need to go and download and ISO, go through each of the install processes, setup network cards, mount and unmount shares etc, Vagrant will automatically do all of this for you. The only requirement is that you must have VirtualBox installed on your local machine.

In this section I will show you how to create the exact machine that we created using VirtualBox but instead we will use Vagrant to automate all of this tasks.

This is great, because now you can just give one of your team members a Vagrant file and they just run one single command `vagrant up` and they will be ready to go.

Before we get started installing vagrant, here is a list of the benefits of using vagrant.

- Automated install of OS's
- Automated network configurations

- Automation with Puppet or Chef
- Cross Platform Development
- Virtualization Manager for Teams

**Vagrant Install:** Grab a copy of the application at the [download](#) page, then install the software on your computer. Vagrant supports OSX, Windows, and Linux distros.

To check if you have successfully setup and configured Vagrant use the following command from your terminal.

---

vagrant

---

```
1 $ ~ vagrant version
```

---

Result

---

```
1 Installed Version: 1.7.4
2 Latest Version: 1.7.4
3
4 You're running an up-to-date version of Vagrant!
```

---

Vagrant has a wide variety of commands available to you, to list all the commands available use the following command from your terminal.

---

vagrant

---

```
1 $ ~ vagrant list-commands
```

---

The most common commands that you might use are the following.

Command	Commands
up	Startups a machine
init	Create vagrant file based on url
provision	Provisions a machine
reload	Restarts a machine
halt	Turns off a machine
destroy	Destroys a machine
ssh	Logging via SSH
status	Shows status of a machine

Now that you are familiar with Vagrant it's time to create a Vagrant file that will store all of the configuration needed for your virtual machine.

First you must select a virtual machine that you will want to init your file with, you can find a list of available virtual machines at [Atlas](#). For this example I will be using [vivid64](#), Atlas will provide you with the init command.

Before you run the command, make sure that you located in the folder where you want save your files that will be used in your application.

```
1 $ mkdir ~/Desktop/app  
2 $ cd ~/Desktop/app  
3 $ vagrant init ubuntu/vivid64
```

Now you will have a file named `vagrantfile`.

Result

---

```
1 $ ~ app vagrant init ubuntu/vivid64  
2 A `Vagrantfile` has been placed in this directory. You are now  
3 ready to `vagrant up` your first virtual environment! Please read  
4 the comments in the Vagrantfile as well as documentation on  
5 `vagrantup.com` for more information on using Vagrant.
```

---

Now that you have created the `Vagrantfile` it's time to startup the machine. If it's your first time running this command it could take sometime to complete, due to Vagrant going out and downloading, and configuring all of the files that it requires.

vagrant

---

```
1 $ vagrant up
```

---

Result

---

```
1 $ vagrant up  
2 Bringing machine 'default' up with 'virtualbox' provider...  
3 ==> default: Box 'ubuntu/vivid64' could not be found. Attempting to fi  
4 tall...  
5     default: Box Provider: virtualbox  
6     default: Box Version: >= 0  
7 ==> default: Loading metadata for box 'ubuntu/vivid64'  
8     default: URL: https://atlas.hashicorp.com/ubuntu/vivid64  
9 ==> default: Adding box 'ubuntu/vivid64' (v20151211.0.0) for provider:  
10    default: Downloading: https://atlas.hashicorp.com/ubuntu/boxes/viv  
11 ons/20151211.0.0/providers/virtualbox.box  
12 ==> default: Successfully added box 'ubuntu/vivid64' (v20151211.0.0) f
```

```
13 lbox'!
14 ==> default: Importing base box 'ubuntu/vivid64'...
15 ==> default: Matching MAC address for NAT networking...
16 ==> default: Checking if box 'ubuntu/vivid64' is up to date...
17 ==> default: Setting the name of the VM: app_default_1450318941149_361
18 ==> default: Clearing any previously set forwarded ports...
19 ==> default: Clearing any previously set network interfaces...
20 ==> default: Preparing network interfaces based on configuration...
21     default: Adapter 1: nat
22 ==> default: Forwarding ports...
23     default: 22 => 2222 (adapter 1)
24 ==> default: Booting VM...
25 ==> default: Waiting for machine to boot. This may take a few minutes.
26     default: SSH address: 127.0.0.1:2222
27     default: SSH username: vagrant
28     default: SSH auth method: private key
29     default: Warning: Connection timeout. Retrying...
30     default:
31     default: Vagrant insecure key detected. Vagrant will automatically
32     default: this with a newly generated keypair for better security.
33     default:
34     default: Inserting generated public key within guest...
35     default: Removing insecure key from the guest if it's present...
36     default: Key inserted! Disconnecting and reconnecting using new SS
37 ==> default: Machine booted and ready!
38 ==> default: Checking for guest additions in VM...
39     default: The guest additions on this VM do not match the installed
40     default: VirtualBox! In most cases this is fine, but in rare cases
41     default: prevent things such as shared folders from working proper
42 see
43     default: shared folder errors, please make sure the guest addition
44 he
45     default: virtual machine match the version of VirtualBox you have
46 on
47     default: your host and reload your VM.
48     default:
49     default: Guest Additions Version: 4.3.34
50     default: VirtualBox Version: 5.0
51 ==> default: Mounting shared folders...
52     default: /vagrant => /Users/rick/Desktop/app
```

---

Now the machine is created and ready to be used over an SSH connection.

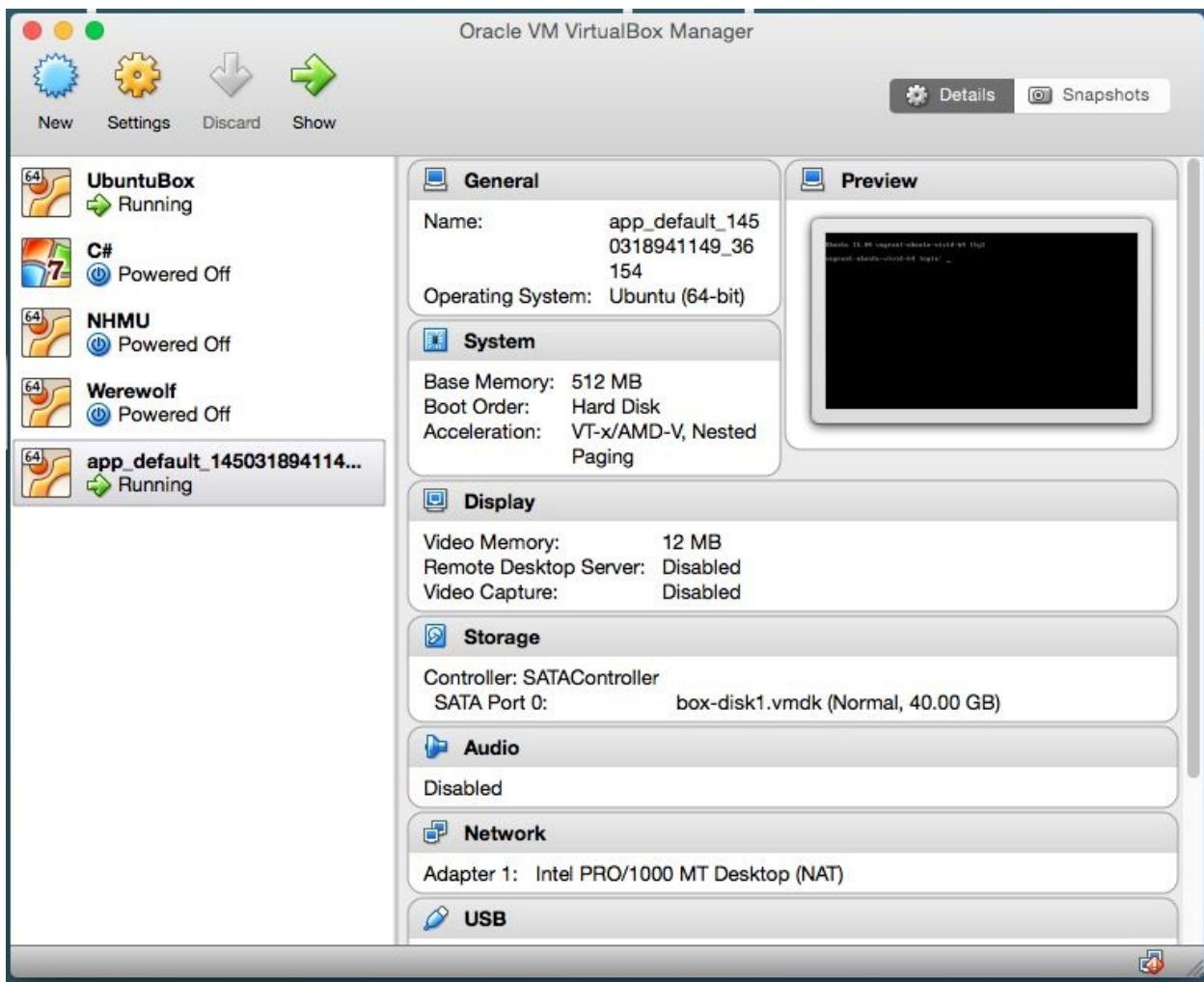
vagrant

---

```
1 $ vagrant ssh
```

---

If you open VirtualBox, you will see the new machine created.



### Vagrant Machine

Not much different then doing it manually, but now you have a shared directory from the local machine and the virtual machine.

When you finish working, you can always shutoff the machine by using the following command

vagrant

```
1 $ vagrant halt
```

Result

```
1 $ vagrant halt
2 ==> default: Attempting graceful shutdown of VM...
```

The last part is to configure the `Vagrantfile` for your specific requirements I.E IP address, port to forward, name, and default shared folder.

Open up the `Vagrantfile` file and uncomment and alter the following lines.

---

#### Vagrantfile Edit

---

```
1 config.vm.network "forwarded_port", guest: 80, host: 3000
2 config.vm.network "private_network", ip: "192.168.56.102"
```

---

Now if you startup the machine you can find it at `192.168.56.102` and the port `3000` being forward. If you would like to change the folder that is being shared you can do this by starting the machine and creating the folder that you want to share.

```
1 vagrant up
2 vagrant ssh
3 mkdir /var/www
4 chown -R vagrant:vagrant /var/www
5 exit
6 vagrant halt
```

Open up the `Vagrantfile` file and uncomment and alter the following lines.

---

#### Vagrantfile

---

```
1 config.vm.synced_folder "./", "/var/www"
```

---

That's it the `/var/www` folder is now being shared. Vagrant has many different functionalities available for you to use. I encourage you to look at the documentation if you are trying to accomplish a very specific task.

[Documentation.](#)

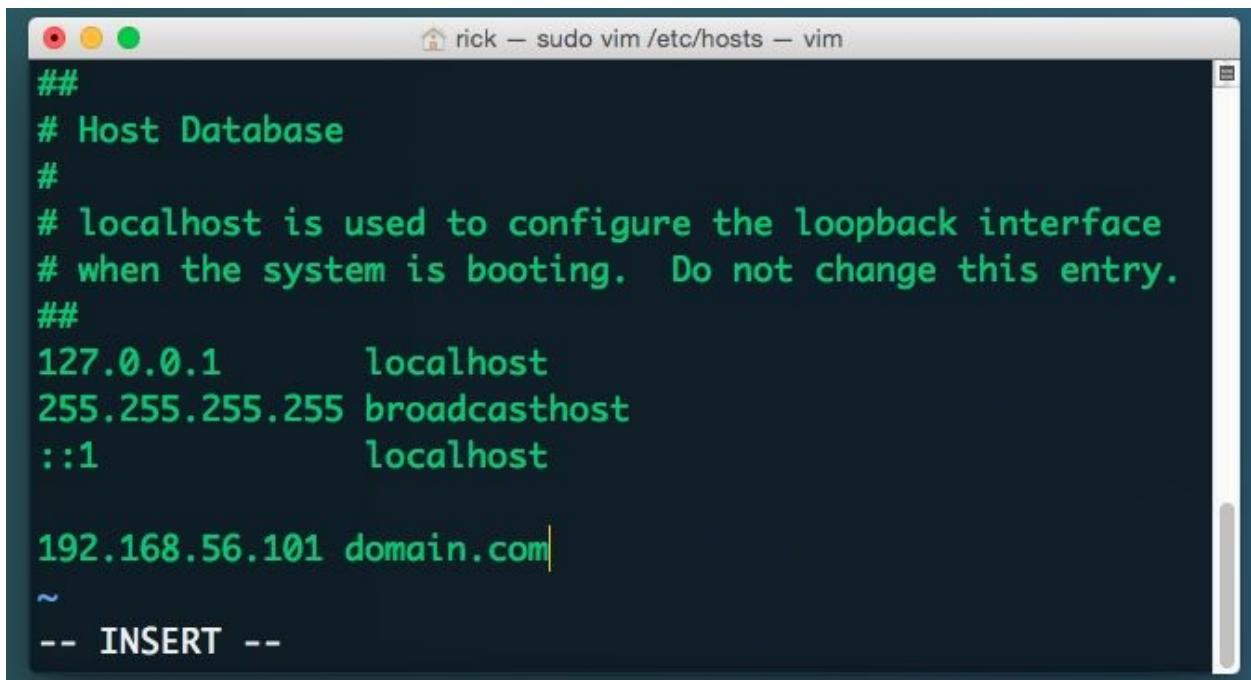
## Custom Hosts File

*This section is completely optional:* The host file that your system uses to match a hostname to an ip address can be located in different locations depending on your operating system.

### Editing the host file

The main purpose for the hostname file is to help your system resolve a name to an address, this is helpful when mounting a share instead of having to memorize an ip address such as `192.168.56.101` you can just use `domain.com` by adding this to the hosts file.

**OSX and Linux** The host file can be found at /etc/hosts, you will have to edit this file with administrative rights.



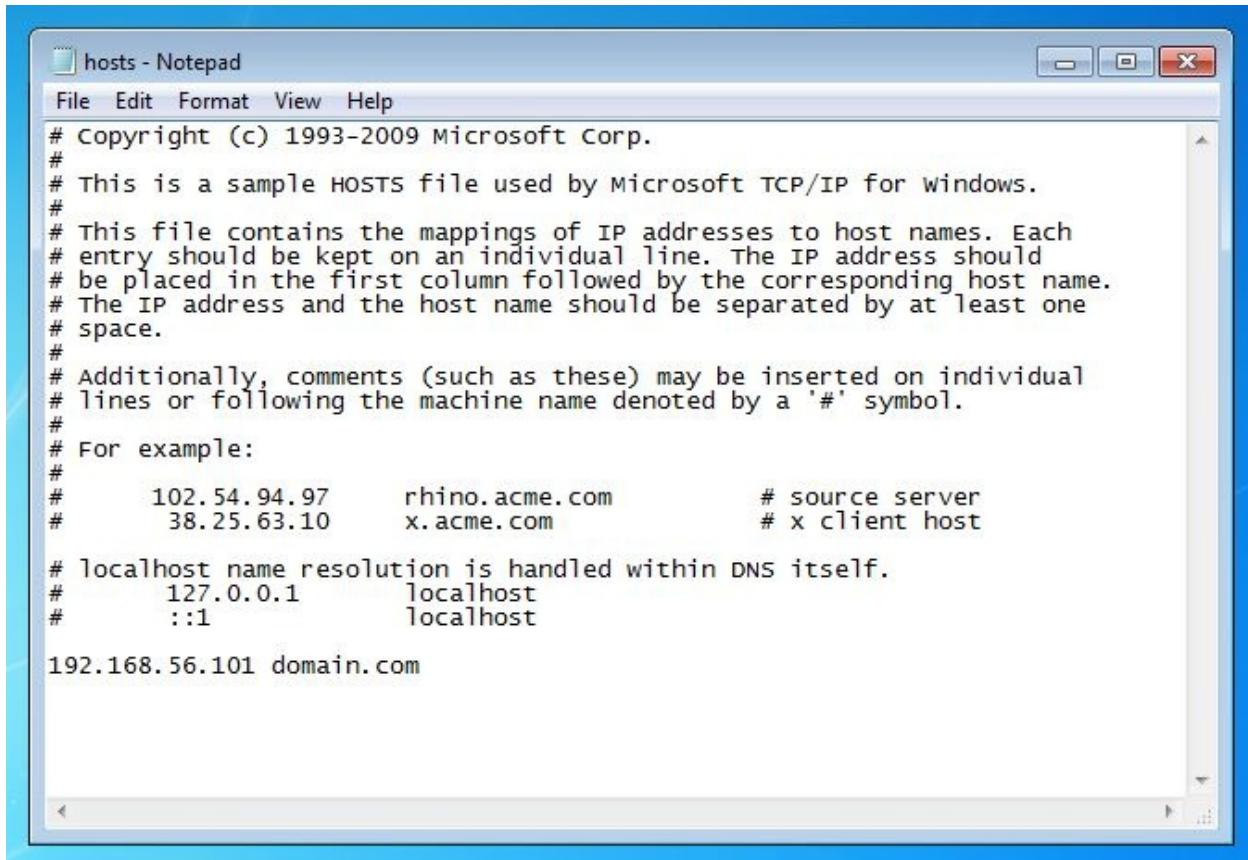
A screenshot of a Mac OS X terminal window titled "rick — sudo vim /etc/hosts — vim". The window shows the contents of the /etc/hosts file in Vim. The file includes standard entries for localhost and broadcasthost, followed by a new entry "192.168.56.101 domain.com" which is currently being typed. Below the file content, the message "-- INSERT --" is displayed, indicating that the user is in insert mode. The terminal window has its characteristic red, yellow, and green close buttons in the top-left corner.

```
##  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1      localhost  
255.255.255.255 broadcasthost  
::1            localhost  
  
192.168.56.101 domain.com  
~  
-- INSERT --
```

Domain Name Host File

**Windows** The host file can be found at

C:\Windows\System32\drivers\etc\hosts, you will have to edit this file with administrative rights.



The screenshot shows a Windows Notepad window titled "hosts - Notepad". The window contains the standard sample hosts file text, which includes comments about the file's purpose, examples of mappings, and localhost entries. At the bottom of the file, there is a single line: "192.168.56.101 domain.com".

```
# Copyright (c) 1993-2009 Microsoft Corp.  
#  
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.  
#  
# This file contains the mappings of IP addresses to host names. Each  
# entry should be kept on an individual line. The IP address should  
# be placed in the first column followed by the corresponding host name.  
# The IP address and the host name should be separated by at least one  
# space.  
#  
# Additionally, comments (such as these) may be inserted on individual  
# lines or following the machine name denoted by a '#' symbol.  
#  
# For example:  
#  
#      102.54.94.97      rhino.acme.com      # source server  
#      38.25.63.10      x.acme.com          # x client host  
  
# localhost name resolution is handled within DNS itself.  
#      127.0.0.1      localhost  
#      ::1            localhost  
  
192.168.56.101 domain.com
```

Domain Name Host File

## Cloud Environment

If your anything like me you probably don't always want to be programming in the same language nor programming in the same machine, you want to see what happens when you run your application in multiple platforms. Cloud Environments allow you just to do that for close to no cost. Cloud environments can be created for as low as \$5 USD a month. In this section we will explore Cloud Environments and find out why and when you should use them.

There is a wide variety of options available for creating cloud development environments, for example Cloud9, Digital Ocean, Codeanywhere, Nitrous and many more. The purpose of this section is to provide you an actionable plan for you to give a cloud environment a try and figure out if it will be a right fit for you or your team.

### Cloud9 Digital Ocean



### Cloud Environment

Meet the IDE that runs entirely on the cloud: [Cloud9](#). This IDE allows developers to be able to work on their code from any machine in any part of the world. All you need is an active connection to the internet, preferably a broadband connection. But why the cloud? First, it enables developers to get started with coding immediately with pre-setup workspaces, collaborate with their peers with shared coding features, and have access to web development features like live preview and browser compatibility testing. It also has a large support for over 40 programming languages including PHP, Ruby, Python, JavaScript with Node.js, and Go.

Here is a list of some of the most popular features:

- Built-in terminal, with npm and basic Unix commands
- Code completion for snippets and identifiers
- Real-time language analysis for JavaScript (marking common JavaScript pitfalls)
- Variable/function name refactoring for JavaScript
- Parenthesis, bracket, and quotation mark character matching
- Line numbers, warnings, and errors in the gutter
- Debugger
- Tabbed file management

- Themes
- Customizable key-bindings, including presets for Vim, Emacs, and Sublime Text
- Code reformatting via JSBeautify and CSSLint
- Supports GitHub, Git, Bitbucket, and FTP servers
- Deployment to Heroku, Joyent, Openshift, Windows Azure, Google App Engine, SFTP/FTP

Will Cloud9 be around? in 2010 the privately held company raised \$5.5 million in Series A funding from Accel Partners and product development software company Atlassian Software. [src](#)

Before getting started create accounts with Cloud9 and Digital Ocean.

- [Create Account with Cloud9](#)
- [Create Account with Digital Ocean](#)

**Why Digital Ocean?** Cloud9 offers virtual machines that you can purchase for a monthly fee. The price ranges start from free to \$79.00 per month. The reason to use Digital Ocean over their environment is that it allows you to have a machine that is constantly running with an SSD drive giving you more control over the environment. This will allow for quick deployment of your application or live debugging with a staging setup since you have complete SSH access to that machine from the IDE.

The first step is to grab the SSH key that Cloud9 provides and let Digital Ocean know about this key. You can find the key at [Link](#)

The screenshot shows the DigitalOcean account settings interface. On the left, there's a sidebar with a user profile picture of a man in an orange shirt, followed by the name "Rick" and "SSH Keys". Below this are menu items: ACCOUNT (Settings, Billing, SSH Keys, Connected Services), and ORGANIZATIONS (Click here for a special surprise). The "SSH Keys" item is highlighted with a blue background. The main content area is titled "Default SSH key" and contains a text input field labeled "KEY VALUE".

### Cloud Key

Digital ocean needs to know about this new key, you can input this key at [Link](#)

## SSH Keys

Add SSH Key

### Add SSH Key

Name  
Cloud9

Public SSH key  
Key Value

Read the guide: How To Use SSH keys With Droplets

Create SSH Key

**Cloud Key**

Now it's time to create a droplet. Start the setup at [Link](#).

# Create Droplet

?

## Droplet Hostname

Cloud9

## Select Size

<b>\$5/mo</b> \$0.007/hour	<b>\$10/mo</b> \$0.015/hour	<b>\$20/mo</b> \$0.030/hour	<b>\$40/mo</b> \$0.060/hour	<b>\$80/mo</b> \$0.119/hour
512 MB / 1 CPU 20 GB SSD Disk 1000 GB Transfer	1 GB / 1 CPU 30 GB SSD Disk 2 TB Transfer	2 GB / 2 CPUs 40 GB SSD Disk 3 TB Transfer	4 GB / 2 CPUs 60 GB SSD Disk 4 TB Transfer	8 GB / 4 CPUs 80 GB SSD Disk 5 TB Transfer
<b>\$160/mo</b> \$0.238/hour	<b>\$320/mo</b> \$0.476/hour	<b>\$480/mo</b> \$0.714/hour	<b>\$640/mo</b> \$0.952/hour	
16 GB / 8 CPUs 160 GB SSD Disk 6 TB Transfer	32 GB / 12 CPUs 320 GB SSD Disk 7 TB Transfer	48 GB / 16 CPUs 480 GB SSD Disk 8 TB Transfer	64 GB / 20 CPUs 640 GB SSD Disk 9 TB Transfer	

## Your Droplet

### Hostname

Cloud9

### Size

\$5/mo

### Region

San Francisco 1

### Image

14.04 x64

### Settings

none

### SSH Keys

none

## Cloud Key

From the selection choose **Node** this will install NodeJS automatically.

## Select Image

Distributions    Applications    Snapshots    Backups

 Dokku v0.4.4 on 14.04	 FreeBSD AMP on 10.1	 Django on 14.04
 LEMP on 14.04	 LAMP on 14.04	 Ruby on Rails on 14.04 (Postgres, Nginx, U...)
 MEAN on 14.04	 Ghost 0.7.1 on 14.04	 Elixir on 14.04
 ELK Logging Stack on 14.04	 Drone on 14.04	 ownCloud 8.2.0 on 14.04
 Joomla! 3.4.5 on 14.04	 Drupal 7.41 on 14.04	 MongoDB 3.0.7 on 14.04
 Redis 3.0.5 on 14.04	 WordPress on 14.04	 Mumble Server (murmur) on 14.04
 PHPMyAdmin on 14.04	 MediaWiki on 14.04	 Cassandra on 14.04
 node v4.2.1 on 14.04	 Docker 1.9.0 on 14.04	 Magento 1.9.2.2 CE on 14.04
 Discourse on 14.04	 GitLab 8.1.3 CE on 14.04	 Redmine on 14.04

### Cloud Key

Make sure to select the right SSH key.

## Available Settings

Private Networking     Backups     IPv6     User Data

### Add SSH Keys (Optional)

C9

+ [Add SSH Key](#)

No root password will be emailed to you because you have selected an SSH Key for access.

[Create Droplet](#)

### Cloud Key

**IMPORTANT** The next step is required in order to move on. You must upgrade your plan to one that allows for SSH support. [Link](#)

## In need of a bigger plan?

Look no further! we have just what you need!

Plan	RAM	Disk Space	Workspaces
Micro	1GB RAM	10GB Disk Space	Max 2 premium workspaces ∞ community workspaces ∞ ssh workspaces
Small	2GB RAM	20GB Disk Space	Max 4 premium workspaces ∞ community workspaces ∞ ssh workspaces
Medium	4GB RAM	40GB Disk Space	Max 8 premium workspaces ∞ community workspaces ∞ ssh workspaces
Large	8GB RAM	80GB Disk	Max 16 premium workspaces ∞ community workspaces ∞ ssh workspaces

**Free** 512MB RAM 1GB Disk Space Max 1 premium workspace ∞ community workspaces Your plan

## Cloud Key

Creating a [workspace](#) requires that it knows about the development machine. This can be accomplished by clicking on the link provided.

## Create a new workspace

Workspace name

Description



Already have your own development machine in the cloud? Create a new cloud9 workspace that connects to it [right here](#).

Hosted workspace

Private

This is a workspace for your eyes only

Public

This will create a workspace for everybody to see

Clone from Git or Mercurial URL (optional)

Choose a template

### Cloud Key

Set up the values for the workspace. Here's what each of the values mean:

- **Hostname:** the domain name or IP address of the machine running your SSH server.
- **Username:** your username for the SSH server (might be root or something else).
- **Node.js Binary Path:** the location of your Node.js binary. If you're not sure where it is, you can always let Cloud9 guess it for you. Otherwise, on your server, type `which node`, to see the full path.
- **Initial path:** this is an optional parameter. You can set it as an absolute (starts with `/`) path. Otherwise, it'll be set to a relative path, and Cloud9 assumes that you want to start from the folder your SSH server leaves you with after login (usually the home or default folder).
- **Port:** this is an optional parameter. Cloud9 will automatically try to connect on port 22.

## Create a new workspace

Workspace name

Description

**SSH workspace** (back to hosted)

### Connect to your server using SSH

Cloud9 will connect to **your server** over SSH using [public key authentication](#).

Please make sure to add our public SSH key to your `.ssh/authorized_hosts` file.

Username

Hostname

Initial path (optional)

Port

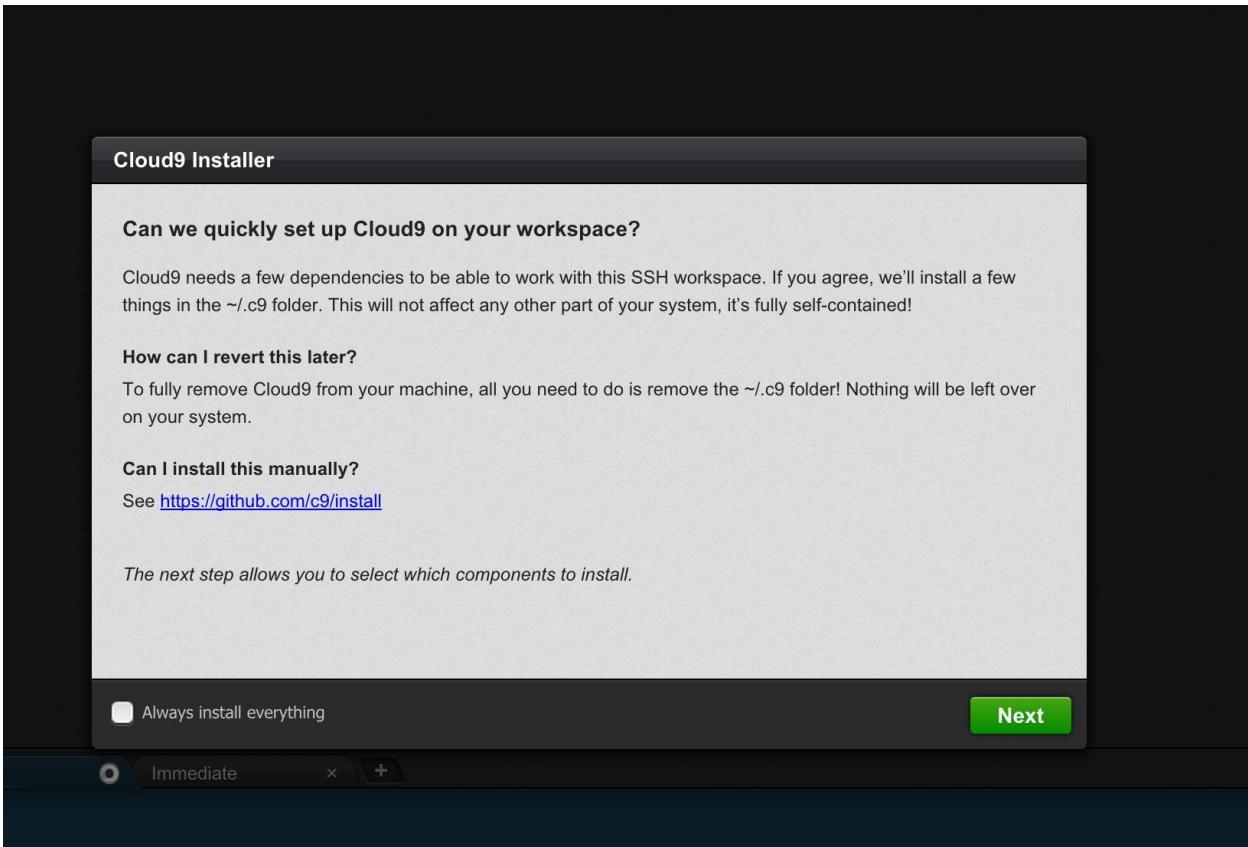
Your SSH key

Node.js binary path (optional)

**Create workspace**

## Cloud Key

Once the workspace is created, you must run a script to initiate the workspace.



## Cloud Key

That's all! Now you should have a working cloud setup to start coding with. If for whatever reason you need to troubleshoot the setup, please use the following resource:

- [Cloud9 Docs](#)



### Pro Tip

Modern HTML5 Editors for example Atom, and Brackets support the ability to be hosted on a remote machine and be used through a web browser.

# Text Editors

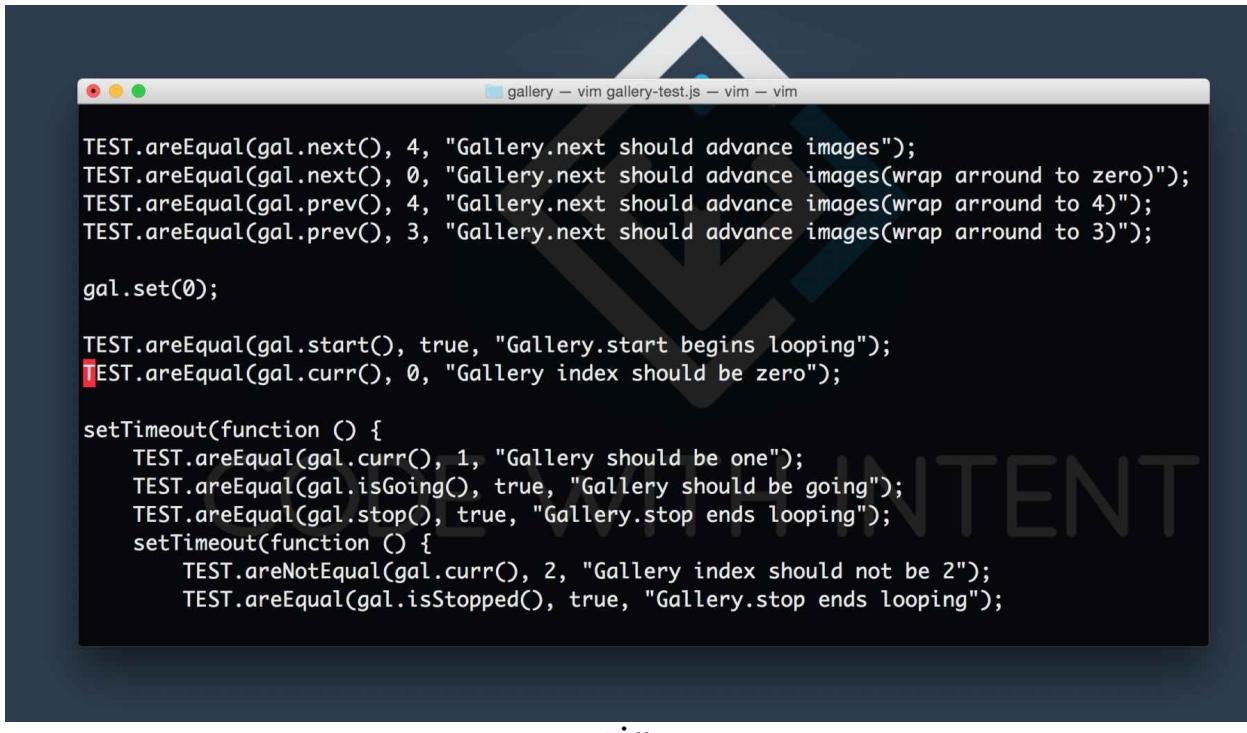
Choosing the right editor is almost as controversial as choosing the right religion. Don't worry; it's not as bad as the internet makes it seem. Choosing the right or wrong editor won't make or break your career, although it still remains an important decision that you must make. Take the following into consideration when choosing an editor:

- Is this editor supported by the community?
- Will this editor be available 5 - 10 years in the future?
- Can I extend this editor with my own plugins?
- Can this text editor help me be more productive?

These are some of the basic questions that you should be asking about your editor. I have compiled a list of editors for you to choose from. The level you are at in your career will determine what editor you decide to use.

**What about an IDE?** Also known as Integrated Development Environments, these types of applications are used to help you create your programs. **I would highly discourage you from using them.** Why? Because you need to be able to understand the tools that are used outside an IDE, such as interpreters (Node.js), build tools (Gulp, Grunt), and package managers (npm). Once you master these tools, you will learn to appreciate an IDE and all its features.

## [Vi / VIM](#)



```
TEST.areEqual(gal.next(), 4, "Gallery.next should advance images");
TEST.areEqual(gal.next(), 0, "Gallery.next should advance images(wrap around to zero)");
TEST.areEqual(gal.prev(), 4, "Gallery.next should advance images(wrap around to 4)");
TEST.areEqual(gal.prev(), 3, "Gallery.next should advance images(wrap around to 3)");

gal.set(0);

TEST.areEqual(gal.start(), true, "Gallery.start begins looping");
TEST.areEqual(gal.curr(), 0, "Gallery index should be zero");

setTimeout(function () {
    TEST.areEqual(gal.curr(), 1, "Gallery should be one");
    TEST.areEqual(gal.isGoing(), true, "Gallery should be going");
    TEST.areEqual(gal.stop(), true, "Gallery.stop ends looping");
    setTimeout(function () {
        TEST.areNotEqual(gal.curr(), 2, "Gallery index should not be 2");
        TEST.areEqual(gal.isStopped(), true, "Gallery.stop ends looping");
    }, 100);
}, 100);

```

vim

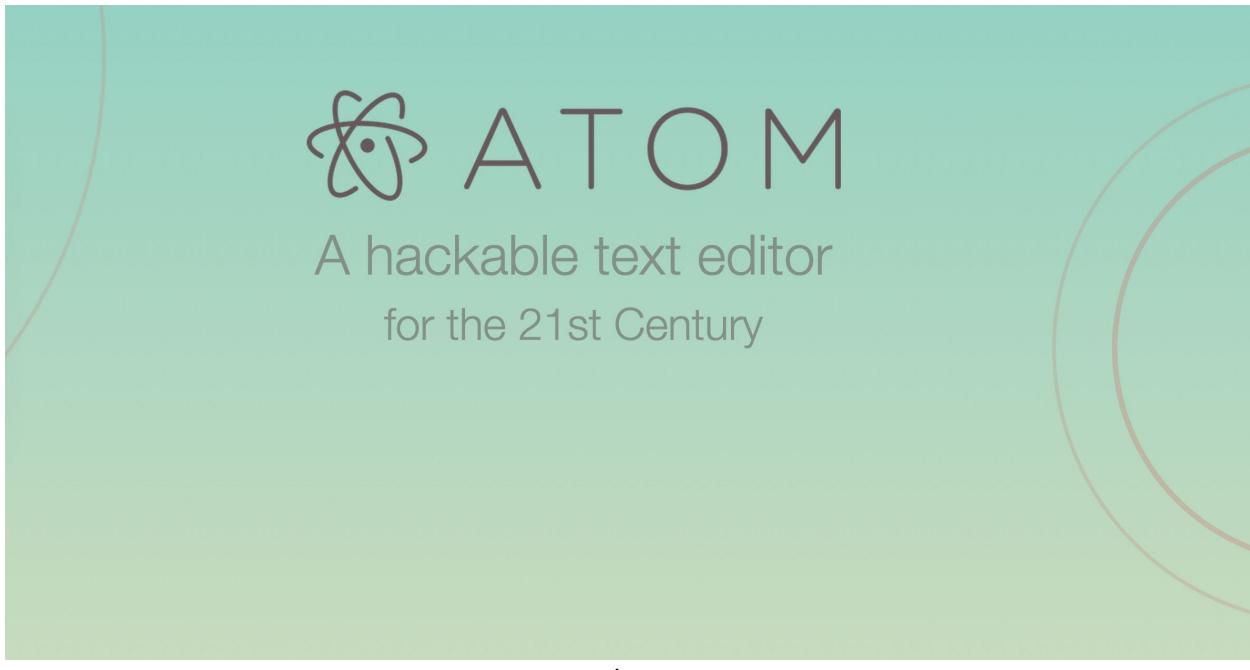
### [Vim is a highly configurable text editor built to enable efficient text editing.](#)

This is probably my favorite text editor. It could be because my education in computer science comes from a university where this was the standard, go-to text editor in the CS department. This text editor comes installed with most modern Unix/Linux distributions. There is no need for instructions, just pull up the terminal and type `vim file.js` and it will work.

#### **Features:**

- Visual Editor
- Built right into the Terminal
- Super Users can code for it in Ruby, Python etc.
- Highly Customizable
- Easily Extensible
- Maintained for 23+ years
- Clear, concise navigation
- Cross platform support
- Free and open source

### [Atom by Github](#)



**atom**

### A hackable text editor for the 21st Century

Atom is a free, open source, and highly customizable text editor. Built around a minimal core, Atom comes with 50 open-source packages plus a large community of developers actively creating and maintaining packages for it. For example, there is a [Vim plugin](#) that you can set up with Atom.

#### **Features:**

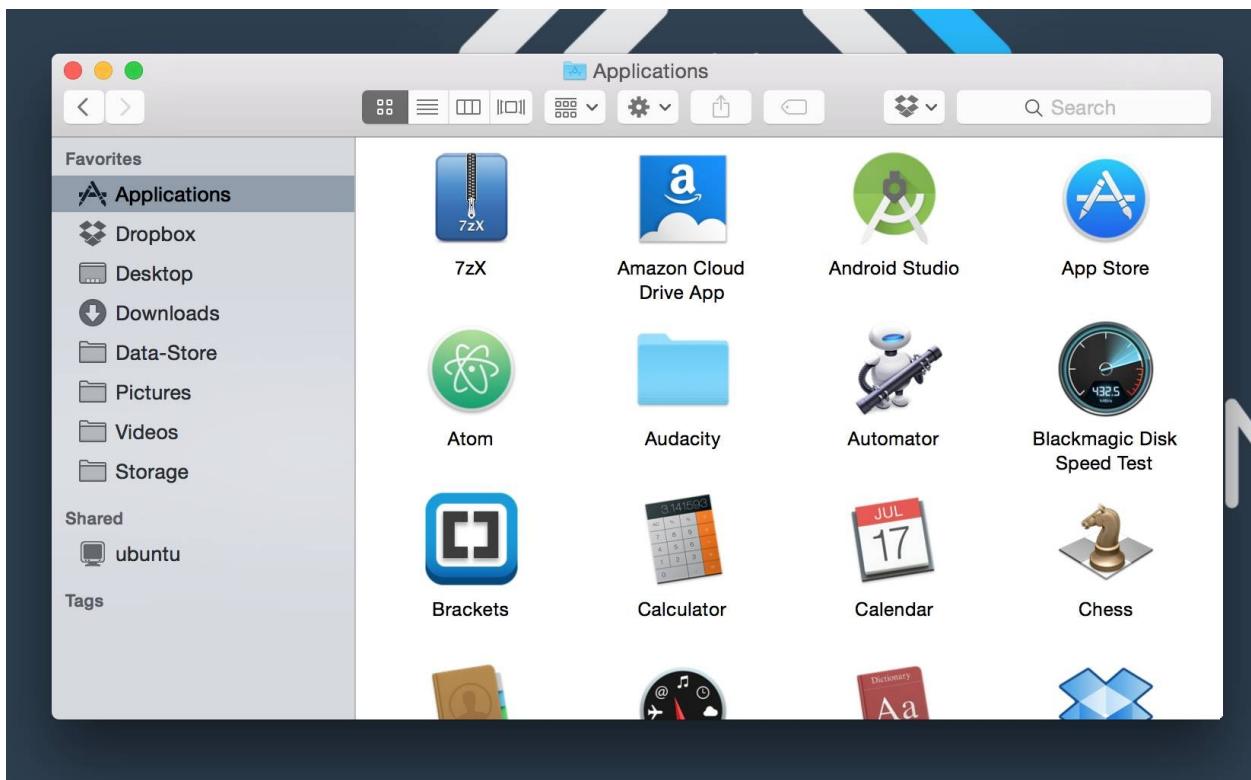
- File system browser
- Fuzzy finder for quickly opening files
- Fast project-wide search and replace (local files only)
- Multiple cursors and selections
- Multiple panes
- Snippets
- Code folding
- Highly extendable
- Maintained for 16 months
- Cross platform support
- Free and open source

Atom is a robust editor that I am sure you will enjoy using for your JavaScript development. Before installing it on your machine, I would suggest you to check out the documentation for the [Atom Flight Manual](#). It will give you a quick guide to getting started with the editor and most importantly on how to develop your own modules for it with native JavaScript and C++.

My personal favorite plugins for Atom: \* [Atom Dark UI theme](#) \* [Solarized Dark UI theme](#) \* [Block Cursor](#) gives you a Vim-like cursor. \* [Mini Map](#) shows a mini map of the entire file. \* [Open Sesame](#) open files with ease. \* [vim-mode](#) provides Vim modal control.

## OSX Setup

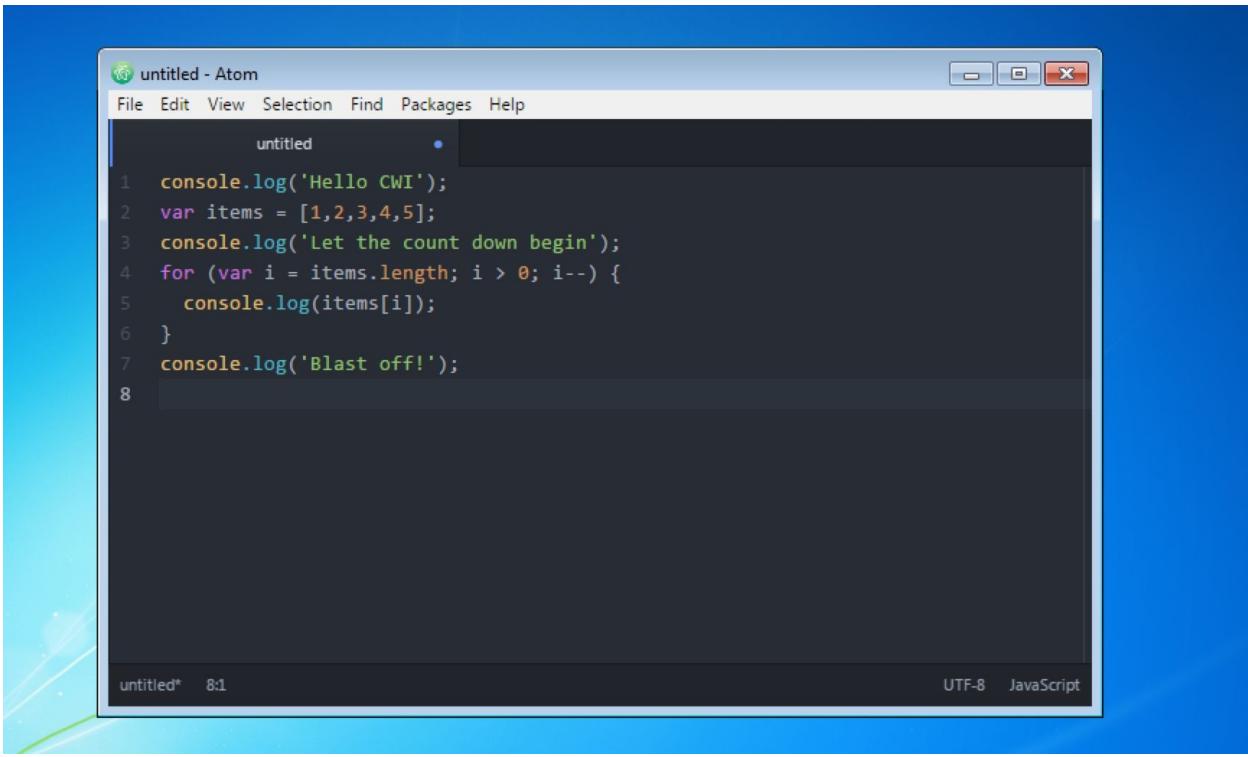
Let's get started by visiting the main website for Atom at [atom.io](#) and selecting the download option. The download will be of a file named `atom-mac.zip`. Extract that file and place the package file into your applications folder.



OSX Setup

## Windows Setup

Let's get started by visiting the main website for Atom at [atom.io](http://atom.io) and selecting the download option. The download will be of a file named `AtomSetup.exe`. Double-click the `AtomSetup.exe` file and wait for Atom to be installed.



A screenshot of the Atom code editor window titled "untitled - Atom". The menu bar includes File, Edit, View, Selection, Find, Packages, and Help. The code editor pane contains the following JavaScript code:

```
1 console.log('Hello CWI');
2 var items = [1,2,3,4,5];
3 console.log('Let the count down begin');
4 for (var i = items.length; i > 0; i--) {
5   console.log(items[i]);
6 }
7 console.log('Blast off!');
8
```

The status bar at the bottom shows "untitled\*" and "8:1" on the left, and "UTF-8 JavaScript" on the right.

## Windows Setup

### Linux Setup

Let's get started by visiting the main website for Atom at [atom.io](https://atom.io) and selecting the download option. The download will be of a file named `atom-amd64.deb` or `atom-amd64.rpm`. Double-click the `atom-amd64.deb` file for Ubuntu and wait for the software center to open. Then, just simply click on *install*, enter your credentials, and wait for Atom to be installed.

A screenshot of the Atom code editor window. The title bar says "hello.js - /home/cwi - Atom". The left sidebar shows a folder icon and "cwi". The main editor area contains the following JavaScript code:

```
1 console.log('Hello CWI');
2 var go = true;
3 console.log('Let the count down begin');
4 while (go) {
5     var count = 10;
6     if (go) {
7         if (count > 0) {
8             console.log(count);
9             count--;
10        } else {
11            go = false;
12        }
13    } else {
14        break;
15    }
16 }
17 console.log('Blast off!');
18
```

The status bar at the bottom shows "hello.js 2:1 (26)" and "UTF-8 JavaScript".

Linux Setup

## Sublime Text

```

base64.cc

31 void base64_encode(const uint8_t * data, size_t length, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < length; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[((s0 & 0xfc) >> 2)];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4));
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < length)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < length) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[((s0 & 0xfc) >> 2)];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4));
54         if (src_idx + 1 < length)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }
57 }

```

Line 31, Column 55      Spaces: 4      C++  
 Use Multiple Selections to rename variables quickly.  
 Here ⌘D is used to select the next occurrence of the current word.

**sublime**

*You'll love the slick user interface, extraordinary features and amazing performance.*

Sublime Text is one of the most widely used text editors among web developers. It's a popular editor due to its extensibility and wide adoption since its original release date of January 18th, 2008; nearly 7 years ago as of this writing. Some of the features are [vintage mode](#), distraction free mode, and quick search.

#### Features:

- Column selection and multi-select editing
- Auto completion
- Syntax highlight and high contrast display
- Macros
- Multiple selections
- Highly extendable
- Maintained for 7+ years
- Cross platform support
- \$70.00 USD and closed source

Sublime Text 2, from my experience, is the go-to editor for web developers. One of the main reasons could be because the editor itself is *fast*. How fast? Well, it's written in native C++ along with Python for most plugins. To really see how fast it is, I would recommend you to test out the "Goto Anything" functionality on a large scale project and see if the results are impressive or not. Aside from its great speed, it has a large amount of [plugins](#) created by developers like yourself. With the low cost of only \$70.00 USD, it's definitely a text editor to keep in the radar for your productivity.

A list of plugins that I have found to be useful:

- [Package Control](#) lets you install community plugins
- [Git](#) enables git support
- [Emmet](#) enables emmet support
- [Vintage Mode](#) enables vim like support
- [Centurion UI theme](#) nice theme

### **OSX Setup**

Let's get started by visiting the main website for Sublime at [sublimetext.com](#) and selecting the download option *Download for OS X*. The download will be of a file named `Sublime Text 2.02.dmg`. Double-click that `.dmg` file and place the package file into your applications folder.

A screenshot of the Sublime Text 2 application window on OS X. The window title is "untitled" and the status bar indicates it is "UNREGISTERED". The code editor contains the following JavaScript code:

```
1 console.log('Hello Everyone!')
2
3 var sublime;
4
5 if (!sublime) {
6     sublime = true;
7 }
8
9 if (sublime) {
10    console.log('Sublime is installed')
11 }
```

The status bar at the bottom shows "Line 10, Column 38", "Tab Size: 4", and "JavaScript".

OSX Setup

## Windows Setup

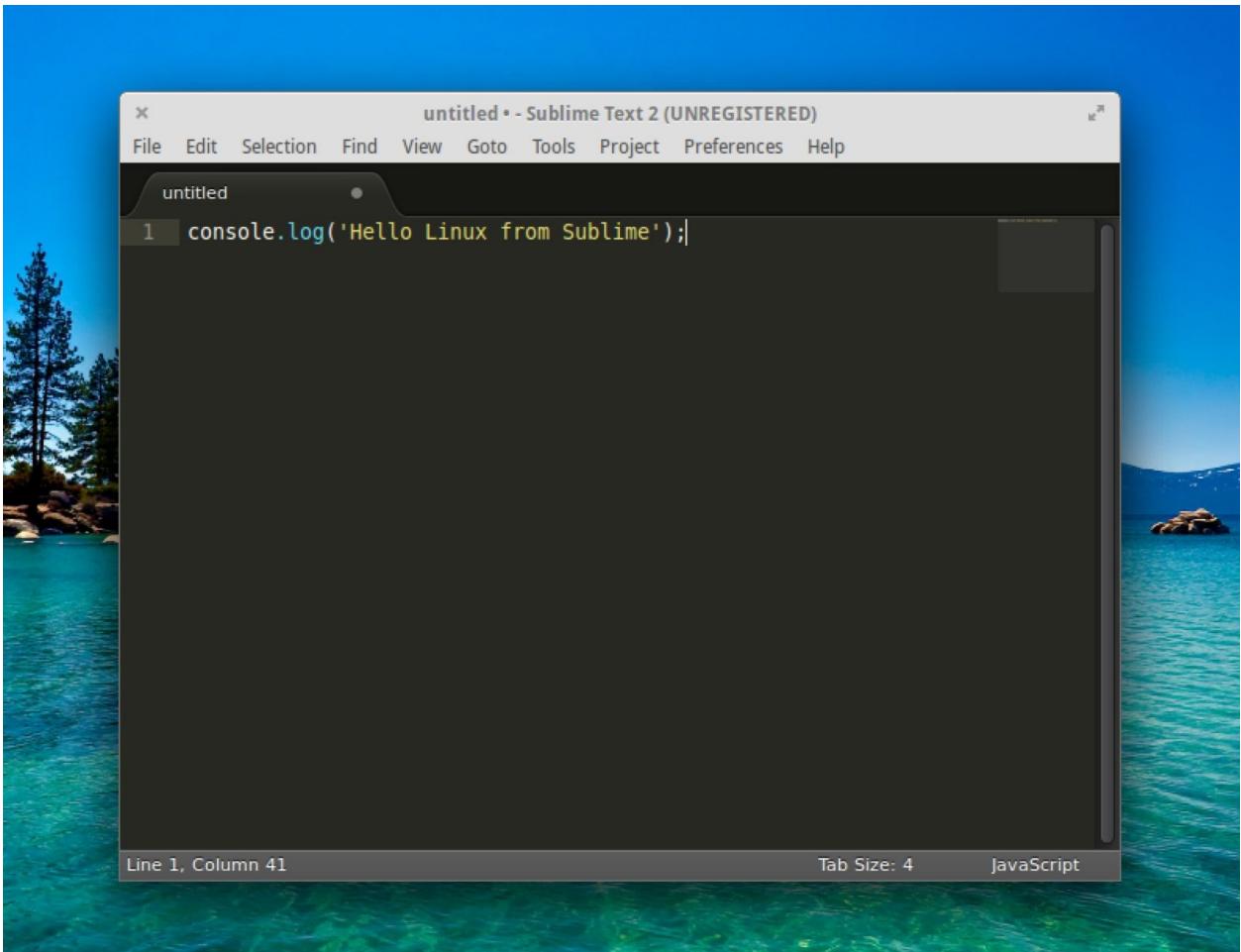
Let's get started by visiting the main website for Sublime at [sublimetext.com](http://sublimetext.com) and selecting the download option *Download for Windows*. The download will be of a file named `Sublime Text 2.02.exe`. Double-click the `.exe` file and follow the instructions. After the install you will have a working Sublime Text editor.

```
C:\Users\Rick\Desktop\web.html - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
web.html x
1 <!--
2
3     [[[[[[[[[[[[[[[      ]]]]]]]]]]]]]]
4     [:::::::::::::      ::::::::::::]
5     [:::::::::::::      ::::::::::::]
6     [::::::[[[[[[[:      :]]]]]]]]:::::
7     [:::::[           ]:::::]
8     [:::::[           ]:::::]
9     [:::::[           ]:::::]
10    [:::::[           ]:::::]
11    [:::::[     CODE THE WEB     ]:::::]
12    [:::::[           ]:::::]
13    [:::::[           ]:::::]
14    [:::::[           ]:::::]
15    [:::::[           ]:::::]
16    [:::::[           ]:::::]
17    [::::::[[[[[[[:      :]]]]]]]]:::::
18    [:::::::::::::      ::::::::::::]
19    [:::::::::::::      ::::::::::::]
20    [[[[[[[[[[[[[[[      ]]]]]]]]]]]]]]
21
22 -->
Line 1, Column 1          Spaces: 4          JavaScript
```

## Windows Setup

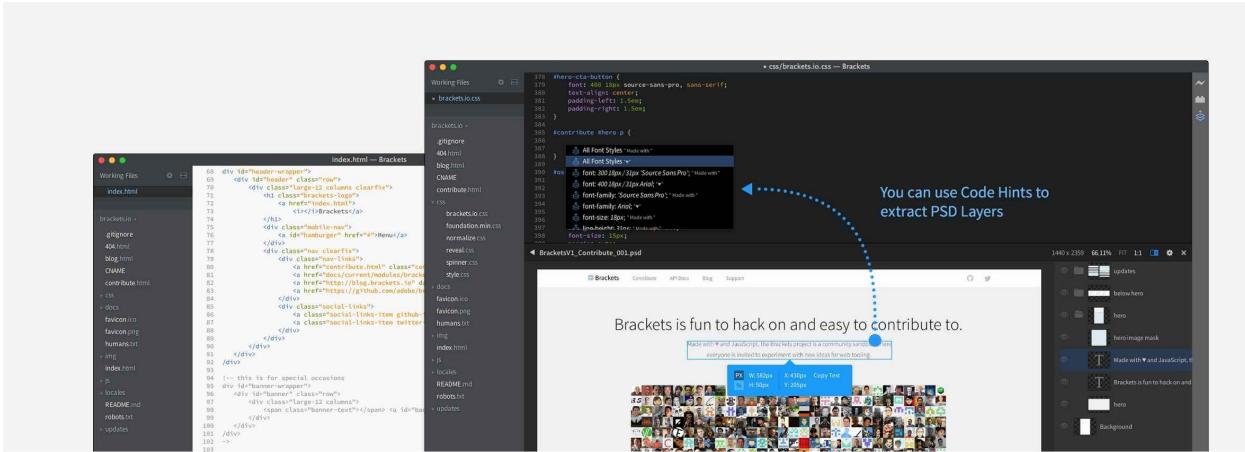
### Linux Setup

Let's get started by visiting the main website for Sublime at [sublimetext.com](http://sublimetext.com) and selecting the download option *Download for Linux*. The download will be of a file named `Sublime Text 2.02.tar.bz2`. Extract the `.tar.bz2` files and place the folder where you have your applications installed. Now just add a symbolic link to the application and you're ready to go.



**Linux Setup**

## [Brackets by Adobe](#)



## Brackets

[A modern, open source text editor that understands web design.](#)

According to Adobe, Brackets is a “A modern, open source text editor that understands web design.” From playing around with Brackets, I have found that it did almost everything I wanted to do. Compared to Atom, Brackets has great support with connecting to remote machines and editing files within those machines.

### Features:

- Live preview
- JavaScript Debugging
- Extract
- Highly extendable
- Maintained for 8 months
- Cross platform support
- Free and open source

As you can see above, this is only a short list of what is available for developers. From my experience, this has been the most interesting project to follow, but of course it's up to you what text editor you decide to use. In this book I will be using Brackets + Vim mode, but any other editor should work fine.

From the first looks at the Atom text editor might have you thinking that it's a text editor for designers. Although this is how the text editor is marketed, don't

be too quick to judge it. Brackets is a lightweight editor built with HTML, CSS and JavaScript with web development as its prime focus. The editor itself is hosted on [github](#) with 22,790 stars. That's 4,186 more than Atom. Not to mention that's its only 8 months old compared to Atom at 16 months!

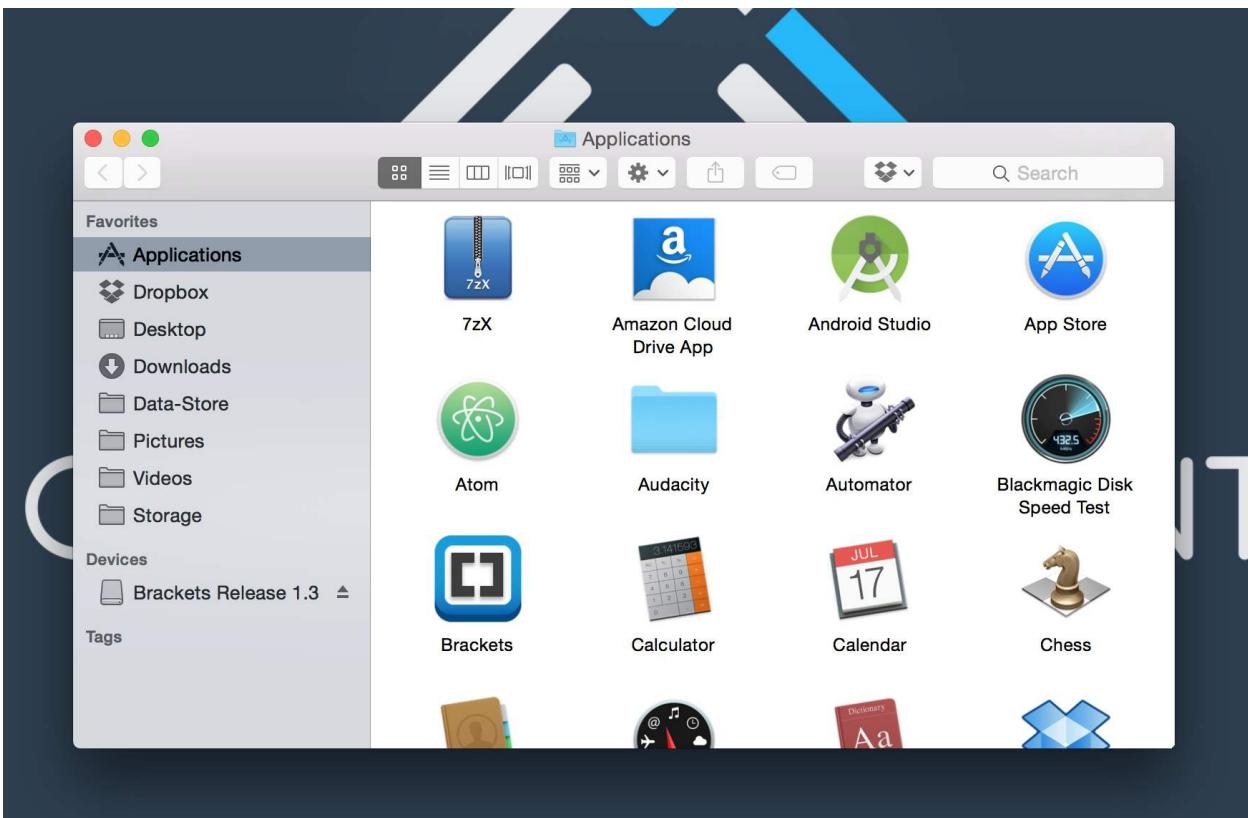
The main reason why Brackets is my current go-to text editor is because it's a tool that does not get in the way of what you want to do with panels, icons, shorthand codes, slow O/I reads and writes. It just does what it supposed to do, which is to help you write more and more every time.

The plugins that I would recommend would be the following:

- [Vimderbar](#) Vim-like functionality
- [Brackets Beautify](#) formats your HTML, CSS, and JavaScript files using js-beautify
- [Git](#) provides git support
- [Icons](#) file icons for tree structure
- [Snippets](#) initiates snippets
- [Document Tool Bar](#) toolbar to view files as tabs
- [Emmet](#) supports Emmet abbreviations
- [File Tree Exclude](#) excludes folders and files from the file tree

## OSX Setup

Let's get started by visiting the main website for Brackets at [brackets.io](#) and selecting the download option *Brackets + Extract(Preview) by Adobe*. The download will be of a file named Brackets.1.3.Extract.dmg. Double-click that .dmg file and place the package file into your applications folder.



OSX Setup

## Windows Setup

Let's get started by visiting the main website for Brackets at [brackets.io](http://brackets.io) and selecting the download option *Brackets + Extract(Preview) by Adobe*. The download will be of a file named `Brackets.1.3.Extract.msi`. Double-click the `.msi` file and follow the instructions. After the install you should have a working Brackets editor.

```
1 <!--
2
3     [::::::[:::::::::::      ]]]]]]]]]]]]]]
4     [::::::::::::::      ::::::::::::]
5     [::::::::::::::      ::::::::::::]
6     [:::::::[[::[:      :]]]]]]]::::::]
7     [::::::[]      ]:::::]
8     [::::::[]      ]:::::]
9     [::::::[]      ]:::::]
10    [::::::[]      ]:::::]
11    [::::::[]      CODE THE WEB      ]:::::]
12    [::::::[]      ]:::::]
13    [::::::[]      ]:::::]
14    [::::::[]      ]:::::]
15    [::::::[]      ]:::::]
16    [::::::[]      ]:::::]
17    [:::::::[[::[:      :]]]]]]]::::::]
18    [::::::::::::::      ::::::::::::]
19    [::::::::::::::      ::::::::::::]
20    [[::::::[::::::::::      ]]]]]]]]]]]]]]
21
22 -->
```

## Windows Setup

### Linux Setup

Let's get started by visiting the main website for Brackets at [brackets.io](http://brackets.io) and selecting the download option *Brackets + Extract(Preview) by Adobe*. The download will be of a file named `Brackets.1.3.Extract.deb`. Double-click the `.deb` file. The Ubuntu software center will open. Now, select the option to `install` and you can begin working with Brackets.

The screenshot shows the Brackets IDE interface. The title bar reads "Untitled-1 (Getting Started) - Brackets". The left sidebar has a "Working Files" section with "index.html" and "Untitled-1" selected. Below it are sections for "Getting Started" and "screenshots", with "index.html" and "main.css" listed under "Getting Started". The main editor area contains a terminal window with the following text:

```
1 <!--
2
3     [:::::::::::::::::::      ::::::::::::::::::::]
4     [::::::::::::::::::::  :::::::::::::::::::::]
5     [::::::::::::::::::::  :::::::::::::::::::::]
6     [::::::::::::::::::::  :::::::::::::::::::::]
7     [::::::[           ]:::::]
8     [::::::[           ]:::::]
9     [::::::[           ]:::::]
10    [::::::[           ]:::::]
11    [::::::[     CODE THE WEB  ]:::::]
12    [::::::[           ]:::::]
13    [::::::[           ]:::::]
14    [::::::[           ]:::::]
15    [::::::[           ]:::::]
16    [::::::[           ]:::::]
17    [::::::[:::::::::::  :]]:::::::::::]:::::::::::]
18    [::::::::::::::::::::  ::::::::::::::::::::]
19    [::::::::::::::::::::  ::::::::::::::::::::]
20    [:::::::::::::::::::  ::::::::::::::::::::]
21
22 -->
23
```

The bottom status bar shows "Line 23, Column 1 — 23 Lines", "INS", "Text ▾", "Spaces: 4", and other icons.

i

## Pro Tip

**Tip:** If you are running ubuntu 15.04 or greater, you need to install the latest version of [libgcrypt11](#)

Linux

# Command Line Tools

Command line (CLI) tools are an essential tool set for any programmer. In this section will be taking a closer look into common command line tools. After completing this section you will have a clear understanding of how and what tools you will need for your “*Perfect Development Environment*”.

Modern JavaScript development can become complex, with larger and larger projects. This complexity is due to having to be able to support development across multiple platforms, with larger and larger teams. Once you start to work on some of this larger programs, you will start to find your self repeating task over and over again. Some of this task could be starting and stopping virtual machines, checking in code to your version control, compilation of source code, source code bundling, source code minification, running tests, starting and stopping node, and many more tasks. This chapter will help you with automation of the most common tasks that you will need when working with JavaScript.

## OSX

I gathered a list of the most popular tools that you will need when developing JavaScript applications. None of this tools are required, but they are a great starting point for JavaScript programmers to have.

**[Homebrew](#)**: Homebrew is a popular package manager for OS X. To install Homebrew open up the terminal and run the following command.

Homebrew Install

---

```
1 ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install)
2 install)"
```

---

**[Git](#)**: Git is a type of version control system designed to handle everything from small to very large projects with speed and efficiency.

**[Oh My Zsh](#)**: Zsh is an open source, community-driven framework for managing your zsh configuration. To install Zsh open up the terminal and run the following command.

## Zsh Install

---

```
1 sh -c "$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)"
```

---

## OSX Automation

In this section you will find some automation tips and tricks that you can use for your workflow in order to make you a more productive programmer.

### Virtual Box and Samba Shares

If you are running a VirtualBox instance from the GUI, and then mounting the samba share. This is costing you time and effort. Automate this with a simple command to startup your workflow and stop your workflow.

First you need to create an alias to startup the virtual machine. Before you run the following command on the terminal make sure that you change the value of <VMNAME> to the name of the VM that you want to startup.

vmwww alias

---

```
1 echo "alias vmwww='VBoxManage startvm \"<VMNAME>\" --type headless'" >> ~/.bashrc
2 _profile
```

---

Now that you can startup the VM you will need a way to mount a samba share and SSH into the server. Change the <USERNAME> value to reflect your username, change <DOMAIN OR IP ADDRESS> to match your domain or ip address, then open your terminal and run the following command. This command will create an alias for www which will create a directory named www and mount your samba shares to it, then SSH you into the virtual machine.

www alias

---

```
1 echo "alias www='mkdir ~/Desktop/www && mount_smbfs //<USERNAME>@<DOMAIN OR IP ADDRESS>/www ~/Desktop/www && ssh <USERNAME>@<DOMAIN OR IP ADDRESS>'" >> ~/.bashrc
2 _profile
```

---

*Neat!* The last part is to be able to turn off the machine, you can do this by creating another alias that un mounts the samba share, deletes the directory, and powers off the virtual machine. First edit the values to reflect your setup and then run this command.

uwww alias

---

```
1 echo "alias uwww='umount ~/Desktop/www && rm -R ~/Desktop/www && VBoxManage controlvm <VMNAME> poweroff" >> ~/.bash_profile
```

---

Don't forget to tell bash about the changes.

Bash Command

---

```
1 source ~/.bash_profile
```

---

At this point you will have the following commands available to you.

- vmwww starts VM
- www mounts shares and SSH's into VM
- uwww unmounts samba share and powers off VM

### Terminal Aliases

The following alias improves the standard ls command by displaying all files included hidden files, in a long listing format that is human readable with ignored groups.

alias

---

```
1 echo "alias ll='ls -lahG'" >> ~/.bash_profile
```

---

Don't forget to tell bash about the changes.

Bash Command

---

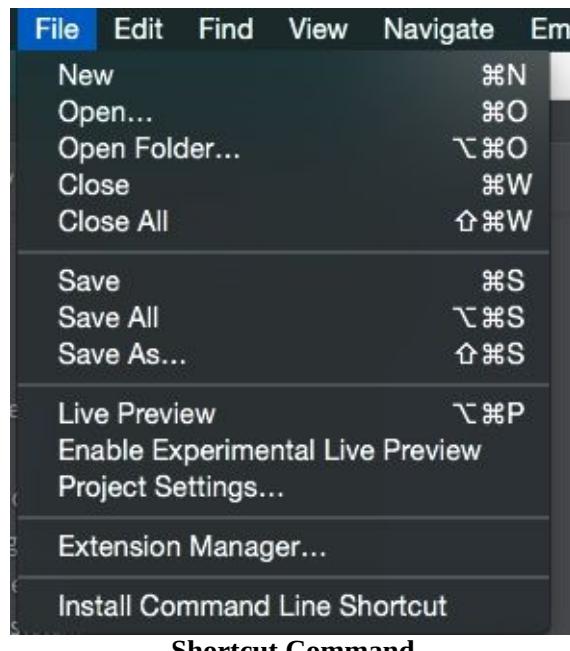
```
1 source ~/.bash_profile
```

---

### Text Editors

At times you might want to open up a file from the command line with your favorite text editor, here I will show you how to accomplish this with Brackets, but this can be modified to work with any text editor.

In Brackets you will need to enable the ability to open the program from the command line by going to file and click on Install Command Line Shortcut in Brackets this will create a symbolic link to the .app file on OSX.



Shortcut Command

Now if you want to open up a file or folder just use the following command.

---

#### Brackets Command

---

```
1 brackets ~/Desktop/app.js
```

---

Here is the full bash script that you can use or modify for your favorite text editor.

---

#### bash script

---

```
1#!/bin/bash
2
3# The actual location of the symlink is passed
4# in $0. We first get the correct target from the
5# symlink and then go about finding the enclosing
6# app. Once a .app is found, we plainly pass the
7# location of the app to open command with a -a
8# as an argument. Note that readlink works differently
9# on mac.
10
11TARGET_FILE=$0
12BRACKETS_APP_NAME="Brackets.app"
13
14TARGET_PATH=$(readlink "$TARGET_FILE")
15TARGET_DIR=$(dirname "$TARGET_PATH")
16FINAL_APP_PATH="$TARGET_DIR"
17if [ ! -z "$TARGET_PATH" -a ! -z "$TARGET_DIR" ]; then
18    BASE_PATH_NAME=$(basename "$TARGET_DIR")
19
```

```

20      # Now that we have found the location go about
21      # trying to find the enclosing app path.
22      while [[ "$BASE_PATH_NAME" != *.app ]]
23      do
24          TARGET_DIR=$(dirname "$TARGET_DIR")
25          BASE_PATH_NAME=$(basename "$TARGET_DIR")
26          if [ -z "$TARGET_DIR" -o "$TARGET_DIR" == "/" ]; then
27              TARGET_DIR="";
28              break;
29          fi
30      done
31      FINAL_APP_PATH="$TARGET_DIR"
32 else
33     FINAL_APP_PATH="";
34 fi
35
36 # Give preference to launching the script's enclosing app
37 # followed by the one in HOME/Applications. Last one
38 # would be checked for global /Applications
39
40 if [ -x "$FINAL_APP_PATH" ]; then
41     open -a "$FINAL_APP_PATH" "$@"
42 elif [ -x "$HOME/Applications/$BRACKETS_APP_NAME" ]; then
43     open -a $HOME/Applications/$BRACKETS_APP_NAME "$@"
44 elif [ -x "/Applications/$BRACKETS_APP_NAME" ]; then
45     open -a /Applications/$BRACKETS_APP_NAME "$@"
46 else
47     open -a Brackets "$@"
48 fi

```

---

## Linux

I gathered a list of the most popular tools that you will need when developing JavaScript applications. None of this tools are required, but they are a great starting point for JavaScript programmers to have.

[Git](#): Git is a type of version control system designed to handle everything from small to very large projects with speed and efficiency.

[Oh My Zsh](#): Zsh is an open source, community-driven framework for managing your zsh configuration. To install Zsh open up the terminal and run the following command.

zsh install

---

```

1 sh -c "$(wget https://raw.github.com/robbyrussell/oh-my-zsh/master/tools/
2 .sh -O -)"

```

---

## Linux Automation

In this section you will find some automation tips and tricks that you can use for your workflow in order to make you a more productive programmer.

### Virtual Box and Samba Shares

If you are running a VirtualBox instance from the GUI, and then mounting the samba share. This is costing you time and effort. Automate this with a simple command to startup your workflow and stop your workflow.

First you need to create an alias to startup the virtual machine. Before you run the following command on the terminal make sure that you change the value of <VMNAME> to the name of the VM that you want to startup.

---

```
alias vmwww
```

---

```
1 echo "alias vmwww='VBoxManage startvm \"<VMNAME>\" --type headless'" >> ~/.bash_\n2 profile
```

---

Now that you can startup the VM you will need a way to mount a samba share and SSH into the server. Change the <USERNAME> value to reflect your username, change <DOMAIN OR IP ADDRESS> to match your domain or ip address, then open your terminal and run the following command. This command will create an alias for www which will create a directory named www and mount your samba shares to it, then SSH you into the virtual machine.

---

```
alias www
```

---

```
1 echo "alias www='mkdir ~/Desktop/www && mount_smbfs //<USERNAME>@<DOMAIN OR IP A\\\n2 DDRESS>/www ~/Desktop/www && ssh <USERNAME>@<DOMAIN OR IP ADDRESS>'" >> ~/.bash_\n3 profile
```

---

*Neat!* The last part is to be able to turn off the machine, you can do this by creating another alias that un mounts the samba share, deletes the directory, and powers off the virtual machine. First edit the values to reflect your setup and then run this command.

---

```
alias uwww
```

---

```
1 echo "alias uwww='umount ~/Desktop/www && rm -R ~/Desktop/www && VBoxManage cont\\\n2 rolvm\n3 '<VMNAME>' poweroff" >> ~/.bash_profile
```

---

Don't forget to tell bash about the changes.

## Bash Command

---

```
1 source ~/.bash_profile
```

---

At this point you will have the following commands available to you.  
\* vwww starts VM  
\* www mounts shares and SSH's into VM  
\* unmounts samba share and powers off VM

## Terminal Aliases

The following alias improves the standard ls command by displaying all files included hidden files, in a long listing format that is human readable with ignored groups.

alias

---

```
1 echo "alias ll='ls -lahG'" >> ~/.bash_profile
```

---

Don't forget to tell bash about the changes.

## Bash Command

---

```
1 source ~/.bash_profile
```

---

## Text Editors

At times you might want to open up a file from the command line with your favorite text editor, here I will show you how to accomplish this with Brackets, but this can be modified to work with any text editor.

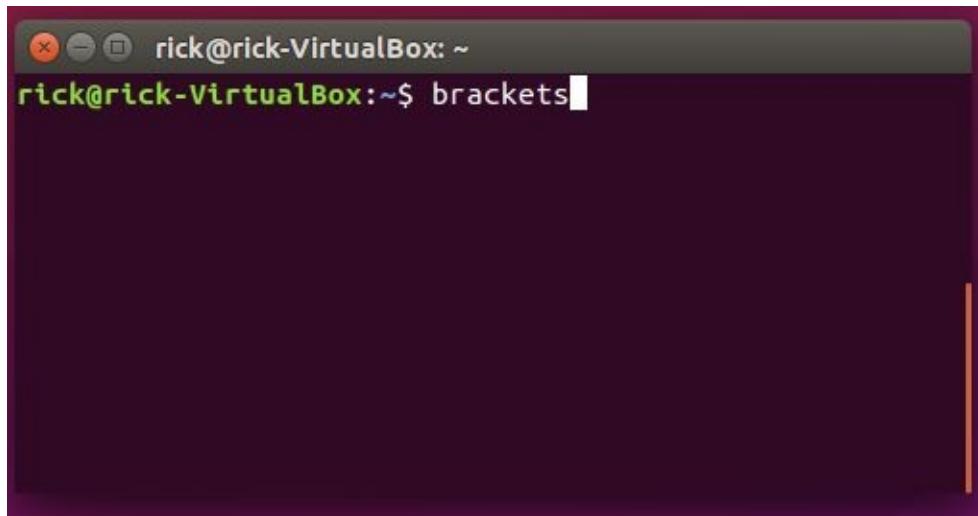
When using a linux distribution and you want to open up a file or folder just use the following command. No need for aliases or shortcuts.

## Brackets Command

---

```
1 brackets ~/Desktop/app.js
```

---

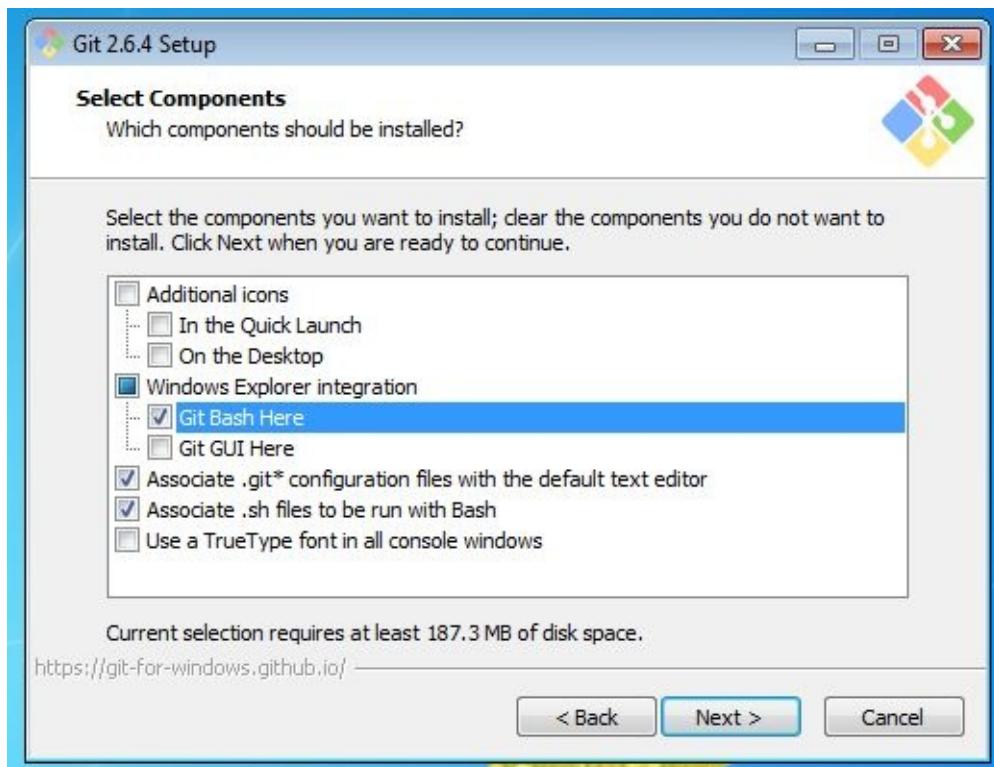


Brackets Ubuntu

## Windows

I gathered a list of the most popular tools that you will need when developing JavaScript applications. None of this tools are required, but they are a great starting point for JavaScript programmers to have.

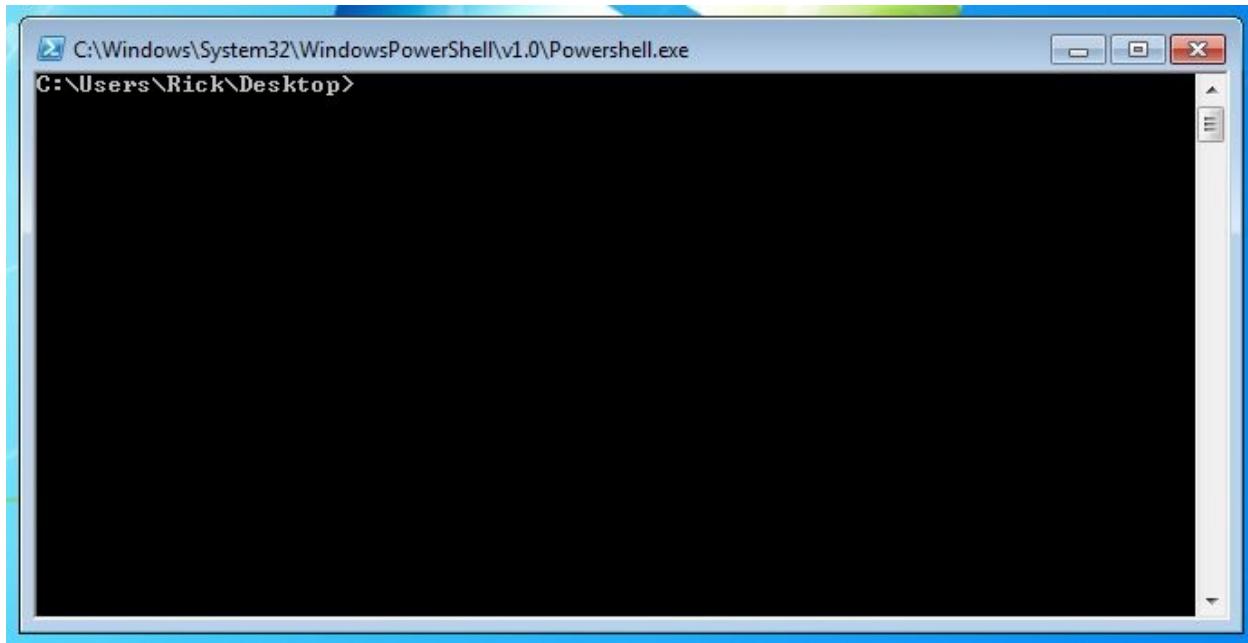
**Git Bash + SSH**: In windows you will need to install **Git Bash + SSH** this will give you a \*nix shell that you can run \*nix commands on, make sure that during the install you setup a shortcut on your desktop in order to access it quickly.



Step 1



Step 2

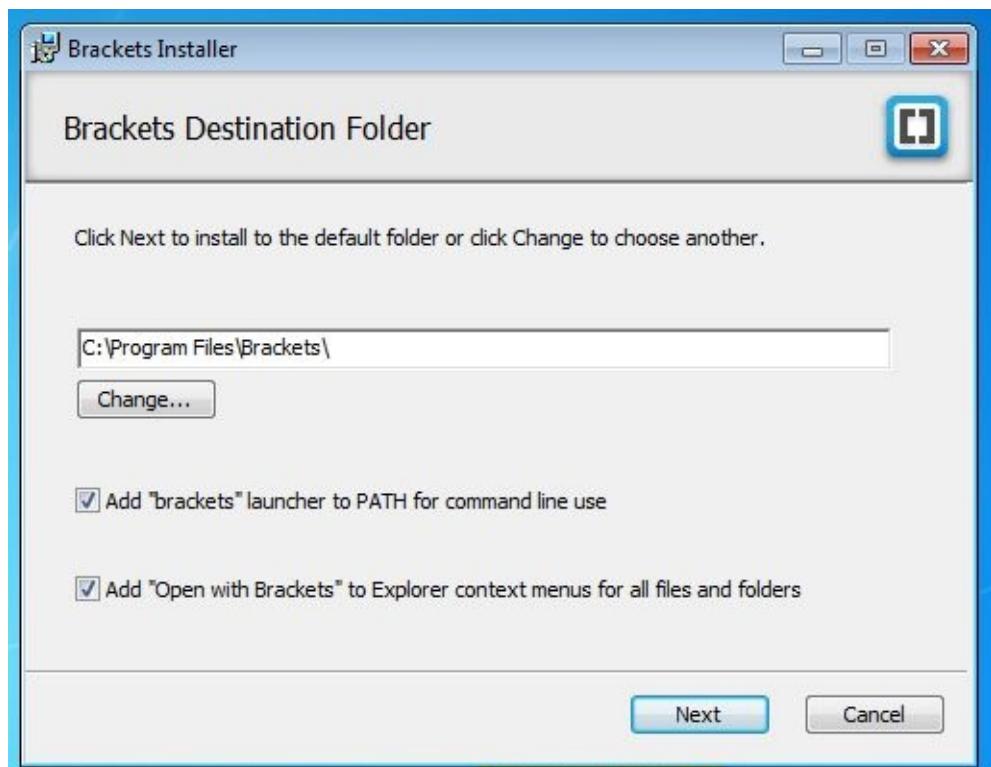


**Step 3**

### **Text Editors**

At times you might want to open up a file from the command line with your favorite text editor, here I will show you how to accomplish this with Brackets, but this can be modified to work with any text editor.

Make sure that when you installed brackets that you selected the Add "brackets" launcher to PATH for command line use option.



Brackets Step 1

The screenshot shows a Windows PowerShell window titled 'C:\Windows\System32\WindowsPowerShell\v1.0\Powershell.exe'. The command entered is 'C:\Users\Rick\Desktop> brackets .\app.js'. The window is mostly black, indicating the command has been run.

Brackets Step 2

# Congratulations!

By reading up to this point you have affirmed to yourself that you are seriously considering going down the path of becoming a programmer. Now you have an idea of what sort of decisions you'll eventually have to make to build up your own personalized workflow and toolset. Keep in mind that as you progress in your new career these decisions will appear time and time again. Your ability to adapt and change your toolset and maintain an efficient workflow will not only make you a much more productive member of a team, but it will also allow you to solve problems from many different approaches.

I want to hear about your development setup, I welcome you to write into [rick@codewithintent.com](mailto:rick@codewithintent.com).

**New programmers from all around the world need your help!** Can you please go onto [jsacademy.com](http://jsacademy.com), [amazon.com](http://amazon.com) or where you purchased this book and leave your feedback about the book. This will help new programmers decide if this book is right for them.

**Leave a review:**

- [JSacademy](#)
- [Amazon](#)

Thank you for allowing me this opportunity of showing you around the world of Development Environments and hopefully you have found the “*Perfect*” Development Environment.

Best wishes,

Rick Hernandez.