



C1 - Dump

C-COD-100

Install Party

Get Ready for Code&Go



- Setup your Ubuntu on VirtualBox
 - Welcome
 - Access to Internet
 - VirtualBox Installation
 - Download Ubuntu
 - Creating a Virtual Machine
 - Setting up your first VM
 - Install Ubuntu 18.04 in VirtualBox
 - Download Ubuntu
 - Ubuntu 18.04 in your VM
 - Guest additions
- Installing Softwares
 - Use terminals
 - About Apt-Get
 - Installing base package
 - Curl
 - Test your installation:
 - Install some text editors
 - SSH
 - Test your installation:
 - PHP
 - MySQL
 - Installation
 - Test your installation
 - Node.JS
 - Definition
 - Test your installation:
 - Python3
 - Definition
 - Test your installation:
 - Ruby on Rails
 - Definition
 - Installation
 - Rails
 - make & Makefile
 - Test
- Cheat Sheet



SETUP YOUR UBUNTU ON VIRTUALBOX

+ WELCOME

Welcome to the Install Party !

This first day marks your first step at EPITECH. Today, you are going to discover how to install Ubuntu on a virtual machine. After that, you will add some useful softwares on your computer. We invite you to do that by yourself and we don't mind if you team up with your peers.

+ ACCESS TO INTERNET

To begin the Ubuntu installation, you'll need an internet connection. You need to follow the next instruction to access to internet.

- Choose **Ionis Portal** network. A popup should appear asking for an authentication.
- Enter the **email and password** we gave to you
- Wait for the redirection.
- If nothing appears, relaunch your browser and repeat point 2.

+ VIRTUALBOX INSTALLATION

First, you'll need to install VirtualBox. Go to [VirtualBox official page](#), and download the latest stable version for windows. (Just click on the "Windows Host" link). Then, just follow the installation. You may need to reconnect to the network during VirtualBox installation.



All links in this document will redirect you to some manual pages to find what you have to do. It's mandatory to do all your research in English.



To change language settings, click on Fichier > Preferences > Language > English (Native).



+ DOWNLOAD UBUNTU

You will need it for later. However, as it might take a while, we **strongly advise you to start it now**.

Please go to [Ubuntu official website](#) and download the 64-bits latest version of Ubuntu 18.04 (Usually named **Ubuntu 18.04.X LTS**).

+ CREATING A VIRTUAL MACHINE

SETTING UP YOUR FIRST VM

Once you've got VirtualBox installed on your computer, please open it. Open it and click on "new" to create a new Virtual Machine, then click on "Expert Mode" on the bottom panel.

Then:

- Choose in the "name" field a name for your VM
- "Type" field must be "Linux"
- "Version" field must be "Ubuntu_64"
- "Memory Size" must be set at 8192MB (If your computer doesn't permit it, set this to a lower value. If your computer has 8GB of RAM, set it at 6144MB)

← Create Virtual Machine

Name and operating system

Name:

Machine Folder:

Type:

Version:

Memory size

4 MB 8192 MB

Hard disk

☐ Do not add a virtual hard disk

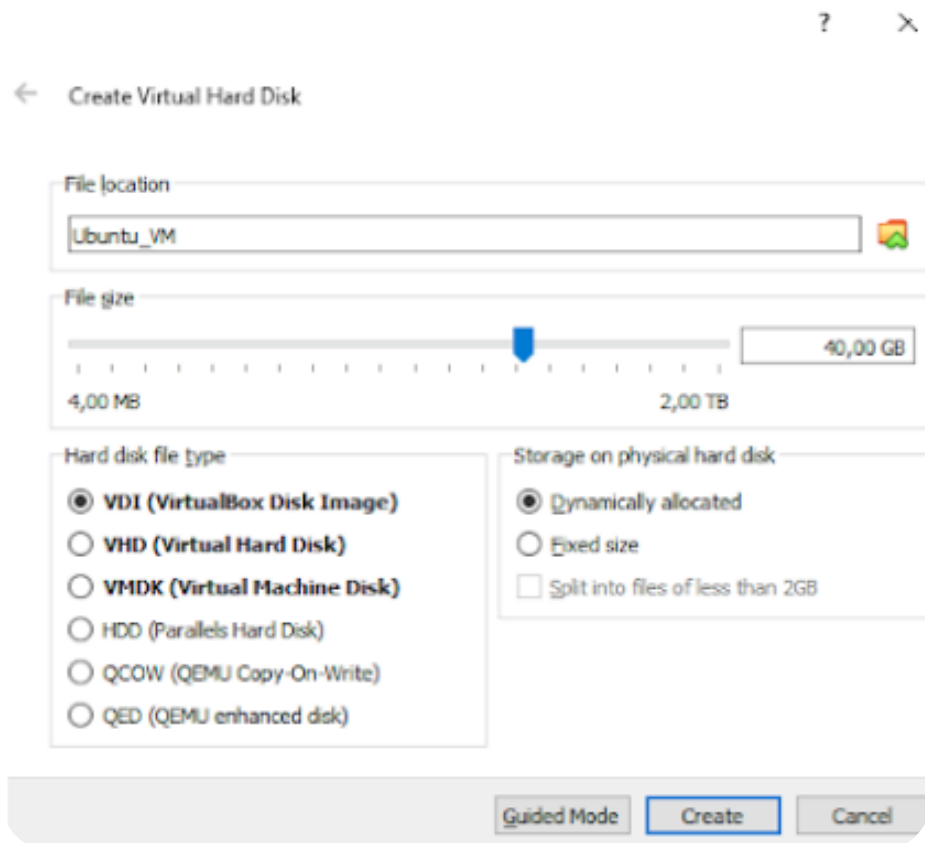
☒ Create a virtual hard disk now

☐ Use an existing virtual hard disk file

Guided Mode Create Cancel



Now, you just have to set your **partition size** (we highly recommend you to set it at least at 40GB, and select option **VDI** (VirtualBox Disk Image), select on the bottom-right panel the option **Dynamically allocated** (very important).

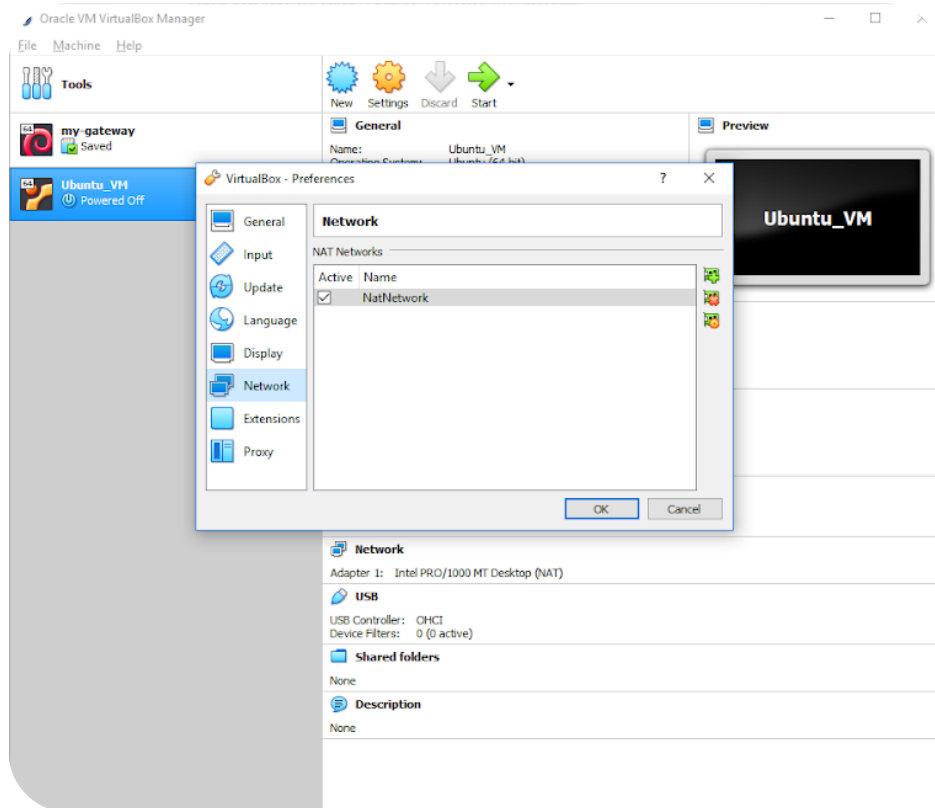


Now click on create and ... congrats ! You have your **first virtual machine** created.



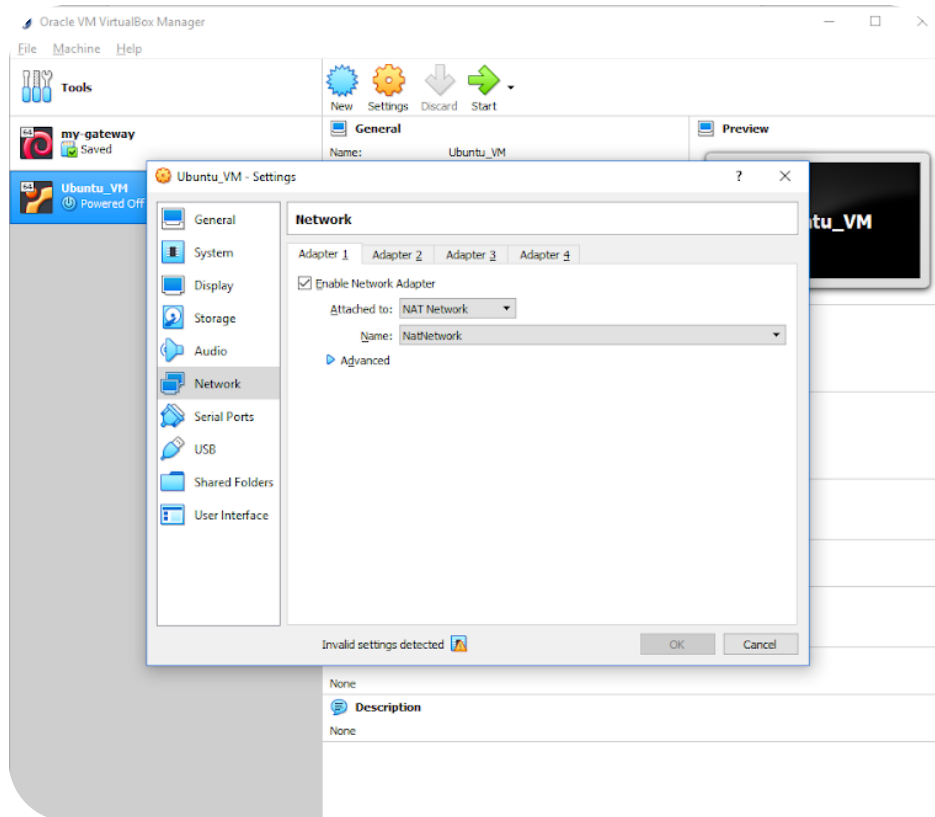
Back to the **VirtualBox** homepage (not the VM's one), please :
open File > Preferences > Network and click on “Add” button on the right-border panel.

This should create a new NatNetwork, like the following screenshot :





Now, go to your **VM settings** by right-click on it, then Settings > Network.
You can now choose the NAT Network type and choose the NAT Network you just created.





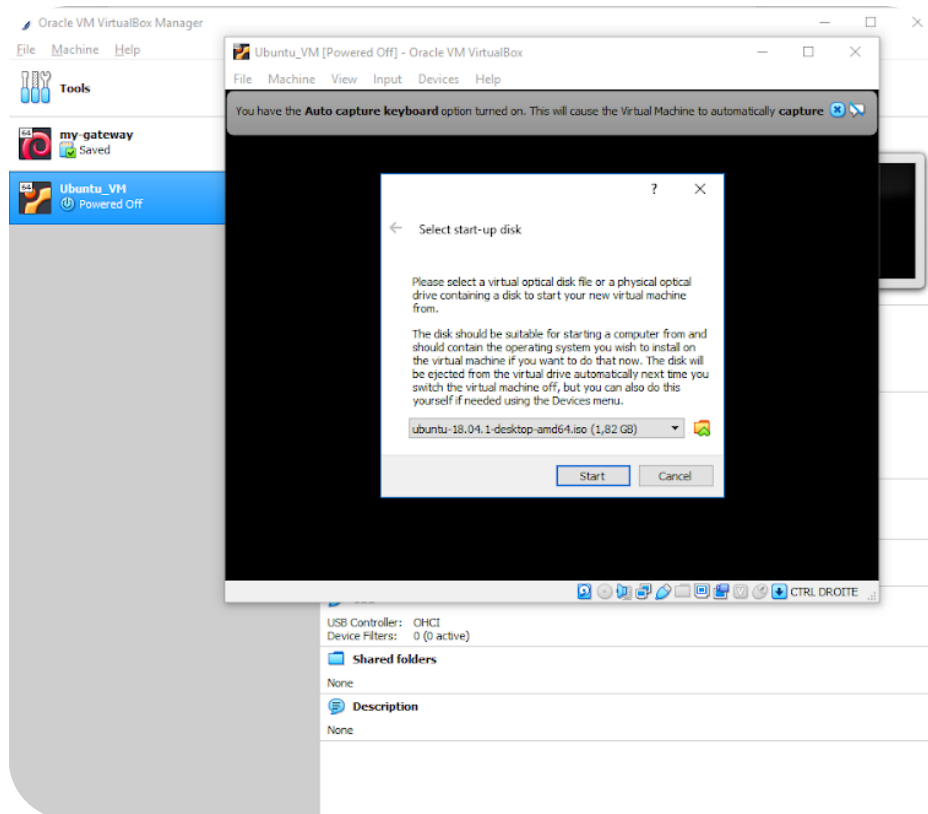
+ INSTALL UBUNTU 18.04 IN VIRTUALBOX

DOWNLOAD UBUNTU

You should already have downloaded Ubuntu. If not, you have now to download an “Disk Image” from the official Ubuntu homepage. Please go to [Ubuntu official website](#) and download the 64-bits latest version of Ubuntu 18.04 (Usually named **Ubuntu 18.04.X LTS**).

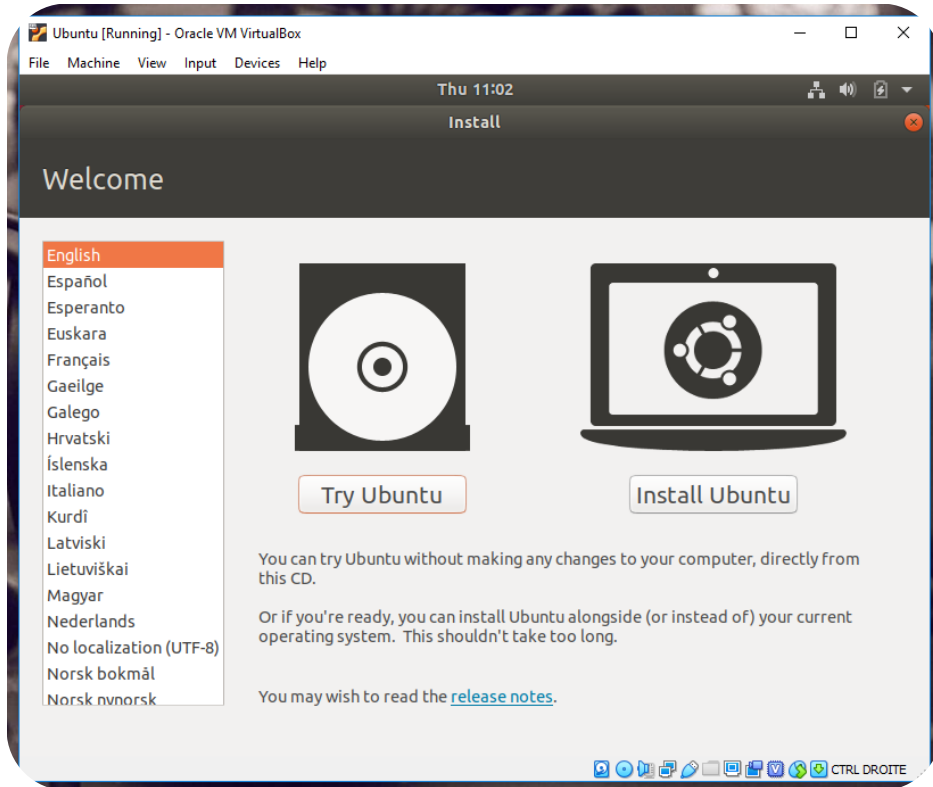
UBUNTU 18.04 IN YOUR VM

Start your virtual machine. VirtualBox should ask you a virtual optical disk file.
Click on Folder button and insert the Ubuntu Image your previously download, then click on **start**.



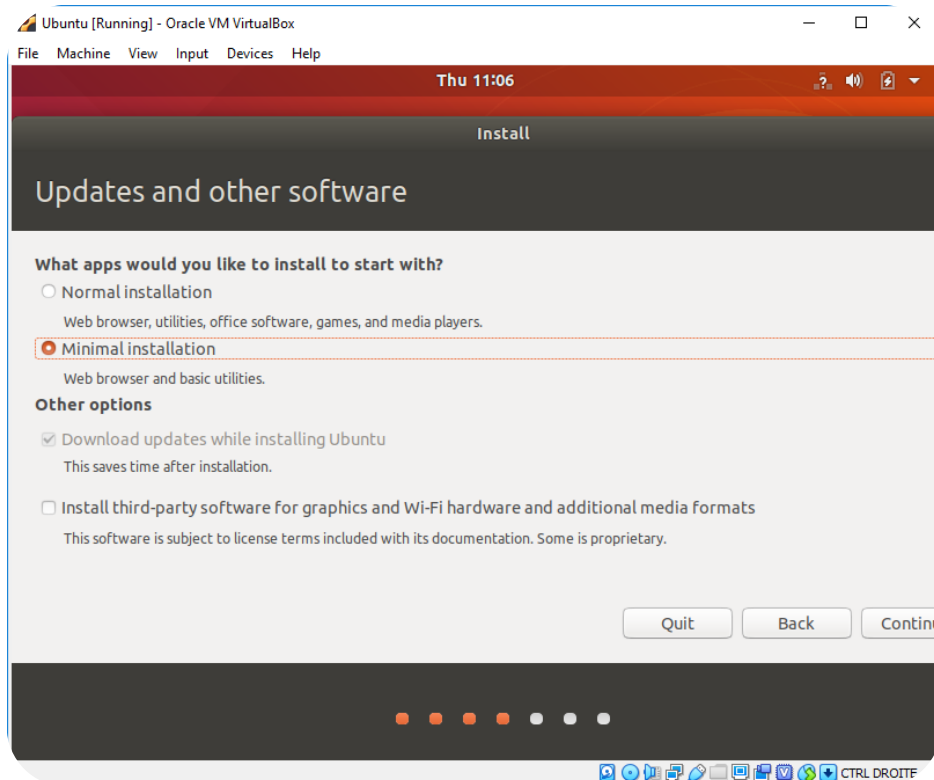
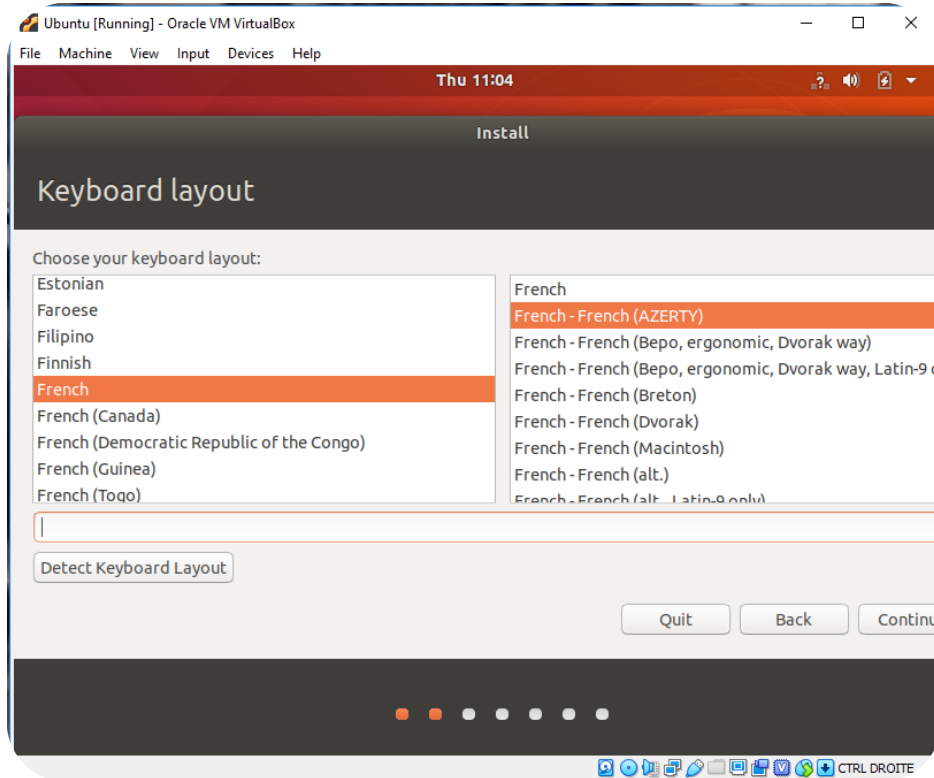


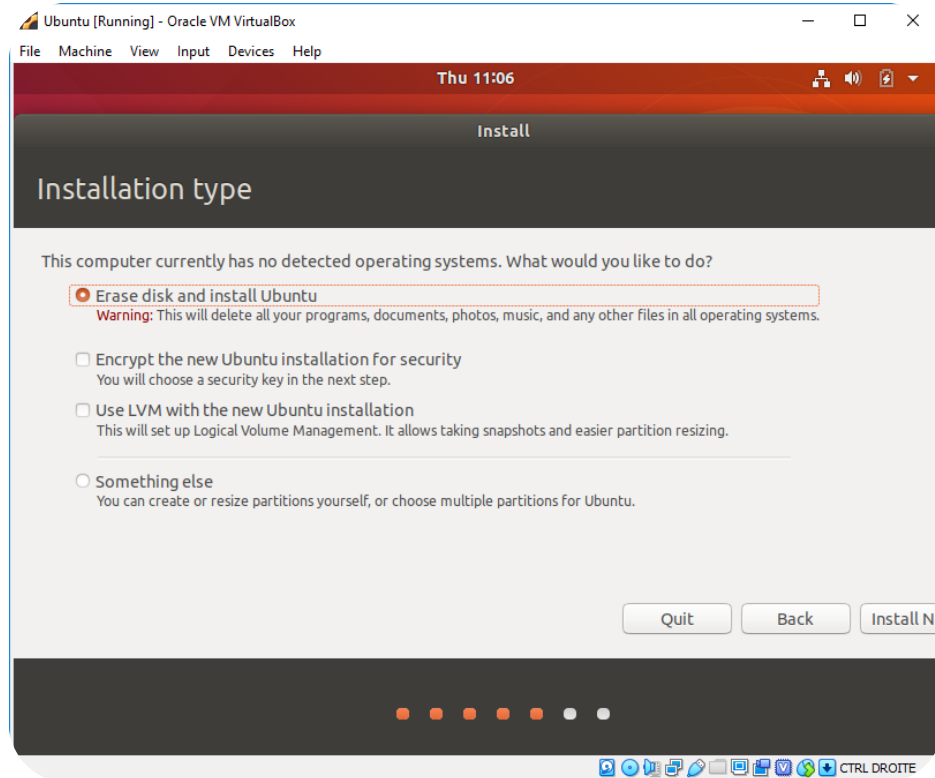
Your VM is starting ! After few minutes, Ubuntu should be launched and should ask you to select a language.



Select “English”, then “Install Ubuntu”.

You can now continue your installation with the following screenshot.





The installation should ask you some personal information, like the name you want to give to your computer, your username, etc...

Feel free to fill it by your own.

When the installation ends, just restart your VM and **you're ready to work !**



You should keep in mind all important information like password you fill during the installation.



You have to keep your personal information for you.



GUEST ADDITIONS

In addition, you can install the guest additions. They permit you to drag and drop files to or from your virtual machine, share the same clipboard between your host machine and your virtual one, etc.

This short [video](#) explains you how to install guest additions.

You can also follow the [VirtualBox manual](#).



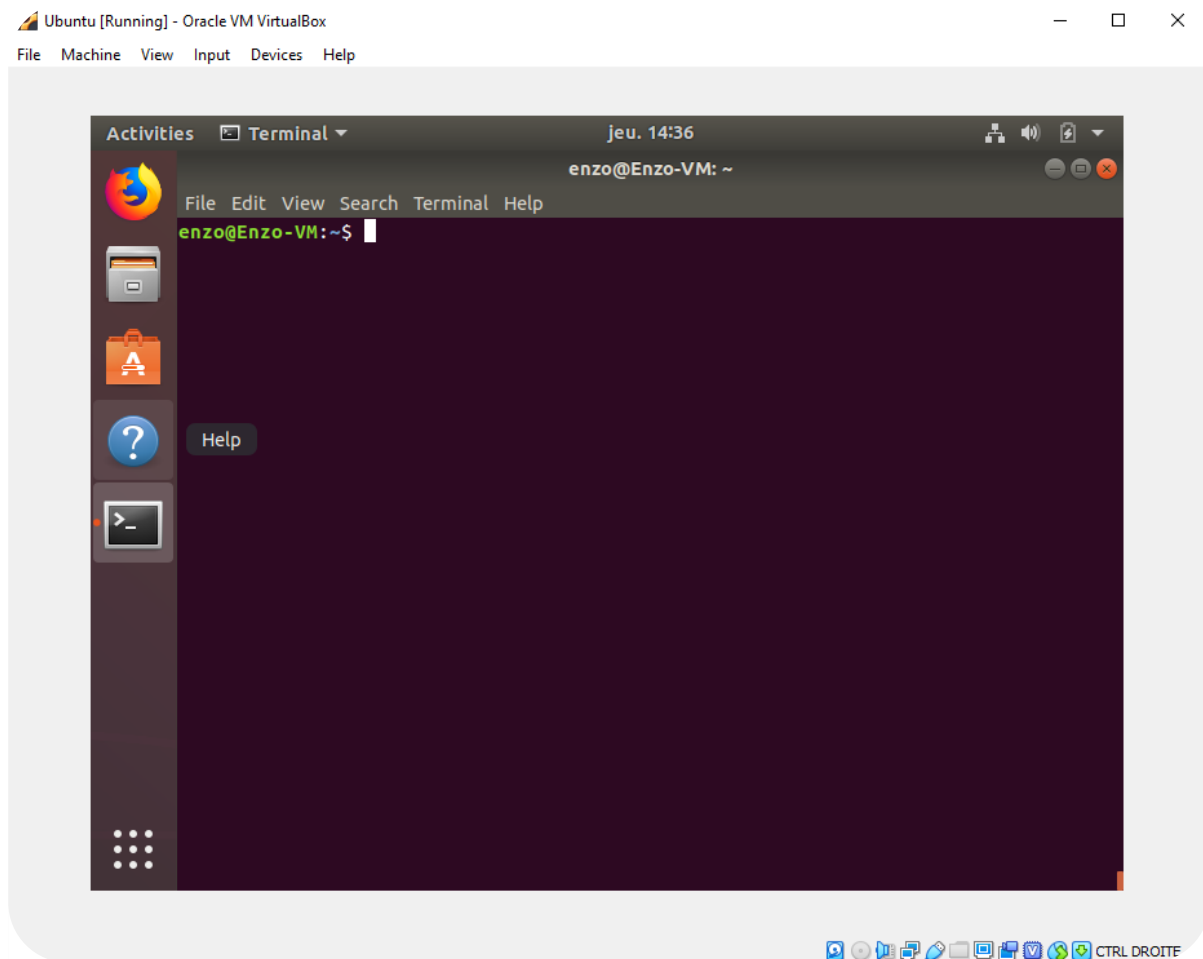
INSTALLING SOFTWARES

+ USE TERMINALS

Now you have your Ubuntu installed on your own VM. We need you to add some software to this OS. In order to do that, you have to open your Ubuntu on your VirtualBox.

In this activity, you will have to open many **terminal** in your Ubuntu. In order to do so, open the **application menu** on the left-bottom corner and select "Terminal" (Or use the command Ctrl + Alt + T).

A **black window** should open, like the following one:



The terminal should be operational by now. If you have some troubles to open it, please ask an assistant.



+ ABOUT APT-GET

apt-get

Advanced Packaging Tool - **APT** - is a package manager used by Debian GNU/Linux and its derivatives. It's a command line tool helpful to install easily a package by name.

Here some usefull commands:

```
Terminal
~/C-COD-100> apt install <package>
~/C-COD-100> apt search <words>
~/C-COD-100> apt remove <package>
~/C-COD-100> apt update
~/C-COD-100> apt upgrade
```

When you are doing an apt operation, get the habit to do an apt update and an apt upgrade before.

This ensures that you will always have the last available packages, and the last version of that tool. First, you need to do an **apt-get update** and an **apt-get upgrade**. The first will update the list of available packages, and the second will upgrade every package on your system with the latest available version.



Every time you want to install a package, you have to **apt-get update** before it.

+ INSTALLING BASE PACKAGE

We are going to install the necessary package for compilation (gcc, g++, etc...). That will be necessary to compile your C files during first week at Coding Academy. Type in your terminal the following command:

```
Terminal
~/C-COD-100> sudo apt-get install -y build-essential libbz2-dev
```



From here, exacts commands will **NOT be provided**. You are encouraged to search by yourself ! Remember, most software can probably be downloaded using apt-get and have a download internet page.



+ CURL



Client URL Request Library is a command line tool used to transfer data using various protocols.

TEST YOUR INSTALLATION:

```
Terminal
~/C-CDD-100> curl -v google.com
* Rebuilt URL to: google.com/
* Hostname was NOT found in DNS cache
* Trying 173.194.45.41...
* Connected to google.com (173.194.45.41) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.38.0
> Host: google.com
> Accept: /
>
< HTTP/1.1 302 Found
< Cache-Control: private
< Content-Type: text/html; charset=UTF-8
< Location: http://www.google.fr/?gfe_rd=cr&ei=22gWVuyzBsHI8gf5x6igCA
< Content-Length: 258
< Date: Thu, 08 Oct 2015 13:00:11 GMT
< Server GFE/2.0 is not blacklisted
< Server: GFE/2.0
<
<HTML>
<TITLE>302 Moved
<H1>302 Moved
....
```

Try for yourself with a cURL request on www.epitech.eu !



If you don't have CURL on your system, please find the apt-get command to install it.



+ INSTALL SOME TEXT EDITORS

Now, it's time to install your first text editors

Default text editors are **nano**, **gedit** and **vi**, but there are others popular ones you have to install:

- **Emacs** is an old terminal text editor. It's arguably better than **vim**. Emacs uses modifier keys to enable shortcuts, which often involves pressing several keys simultaneously.



In C pool you will use emacs to edit code in Shell and C language

- **Vim** is an old terminal text editor. It's arguably better than Emacs. Vim uses editing modes and aims to minimize the number of keystrokes the user has to use.
- **Sublime Text** is a graphical lightweight text editor.
- **Visual Studio Code** is a very good IDE.

There are many others. Thus, you might want to try some IDE (Integrated Development Environment) as they include many useful tools.



+ SSH



Secure Shell (SSH) is a computer program and a secured communication protocol. You are going to use for the first time apt-get to install it on your Linux machine.

TEST YOUR INSTALLATION:

```
Terminal
~/C-COD-100> ssh your_login@localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is 11:7f:e6:31:96:46:1f:26:a2:14:bf:00:db:d4:20:71.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
your_login@localhost's password:
The programs included with the Debian GNU/Linux system are free software; the
exact distribution terms for each program are described in the individual files in
/usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to
the extent permitted by applicable law. No mail.
Last login: Thu Oct 8 13:56:54 2015 from localhost
```

Then, you'll have to **generate your public/privates keys** (RSA 4096).



If you don't have SSH on your system, please find the apt-get command to install it.



+ PHP



PHP is a **server-side** programming language. You need Apache to run PHP file. Please run:

```
Terminal
~/C-COD-100> sudo apt-get install apache2 apache2-doc
```

The PHP configuration used in command line is stored in **/etc/php7/cli/php.ini**.

This is not to be confused with the Apache configuration which is stored in **/etc/php7/apache2/php.ini**.

This is not mandatory, but here are some suggestions for modifications:

```
Terminal
~/C-COD-100> cat /etc/php7/apache2/php.ini
max_execution_time = 30
max_input_time = 60
memory_limit = 64M
upload_max_filesize = 10M
register_globals = Off
expose_php = Off
```



Please refer to the documentation for a complete description.

Don't forget to restart your Apache server after any modification of the PHP configuration. To restart your apache server, you can launch on your terminal this command line:

```
Terminal
~/C-COD-100> sudo systemctl restart apache2
```

In order to test your installation of PHP, we are going to create a PHP file that we will launch in the **CLI** (Command Line Interface) and with Apache.


A default site is created when installing Apache, it is located in **/var/www/html**



Create a file named **index.php** with the following content:


```
Terminal
~/C-COD-100> cat /var/www/html/index.php
<?php
    echo phpinfo();
?>
```

Now if you go on the address **localhost/index.php** you should see the following page:

PHP Version 5.6.13-0+deb8u1

System	Linux jolivad-debian8-vm 3.16.0-4-686-pae #1 SMP Debian 3.16.7-ckt11-1+deb8u4 (2015-09-19) i686
Build Date	Sep 8 2015 12:02:11
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-readline.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.6.0, Copyright (c) 1998-2015 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2015, by Zend Technologies





The **phpinfo()** function is commonly used to test if php works fine.

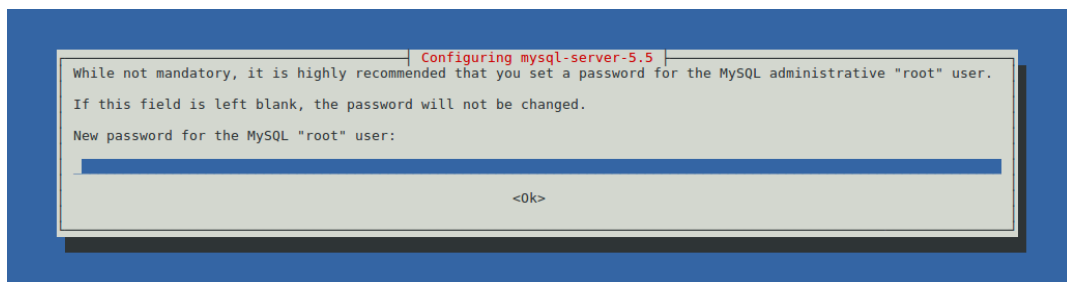


+ MYSQL



INSTALLATION

During the installation process, you should see the following screen asking for a new root password.



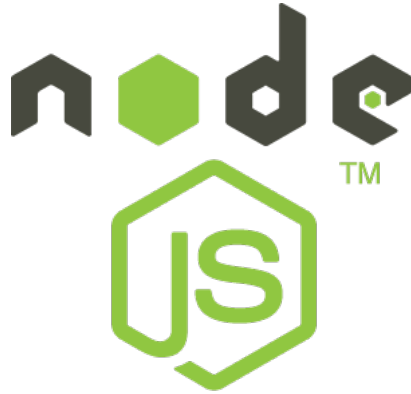
Set the password you want. But **remember it**. After this, your MySQL installation is finished.

TEST YOUR INSTALLATION

```
Terminal
~/C-COD-100> mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ;
Your MySQL connection id is 44
Server version: 5.5.44-0+deb8u1 (Debian)
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other
names may be trademarks of their respective owners.
Type 'help;' for help.
mysql> exit
```



+ NODE.JS



DEFINITION

Node.js is an open source, event-driven software platform in JavaScript designed for scalable network applications. The difference with Apache is that Node is asynchronous, and is on one thread only. But it does the job very quickly!



If you want to launch some .js files in CLI, you will need node.js

TEST YOUR INSTALLATION:

```
Terminal
~/C-CDD-100> node -v
v12.3.1
```



if node -v command line works, it's all good.

To try your new Node.js installation out, create a small **web server**.

Create a folder named nodejs in your home:

```
Terminal
~/C-CDD-100> mkdir /home/your_login/nodejs
```



Then, create a file named `server.js` in this folder:

```
Terminal
~/C-COD-100> cat /home/your_login/nodejs/server.js
var http = require('http');

var server = http.createServer(function(request, response) {
  console.log('New request');
  response.writeHead(200, {'Content-Type' : 'text/plain'});
  response.end('Hello world !');
});

server.listen(3000);
console.log('Server listening on port 3000');
```

Finally, start the server:

```
Terminal
~/C-COD-100> node server.js
```

Go on your browser and type **localhost:3000** in the search bar, and watch what happens.



+ PYTHON3



DEFINITION

Python is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a “batteries included” language due to its comprehensive standard library.

TEST YOUR INSTALLATION:

```
Terminal
~/C-COD-100> python3
Python 3.7.0 (default, Oct 2 2018, 09:18:58)
[Clang 10.0.0 (clang-1000.11.45.2)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
```



If python3 command line works, it's all good.



Press CTRL+D to quit python3



+ RUBY ON RAILS



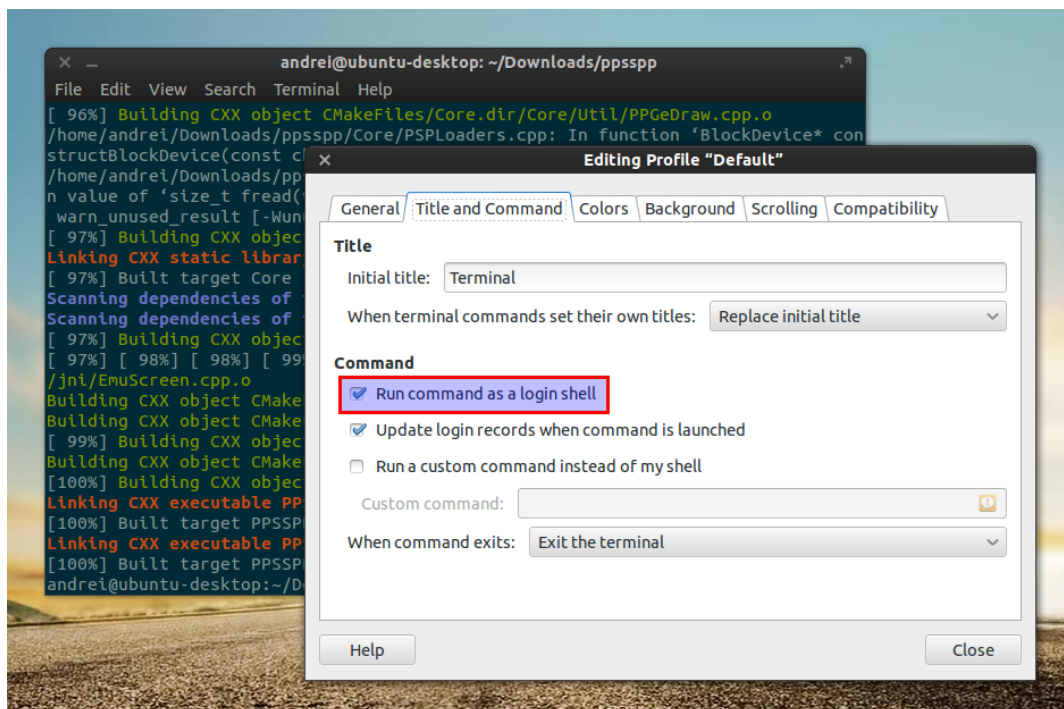
DEFINITION

The Ruby on Rails framework, developed in Ruby, integrates a web server designed to be a development web server which should never be used past development. Thus, when you are ready to go into production with your Ruby masterpiece, you will need to install it on a web server. By default, Apache 2 does not communicate with Ruby, even less so with Ruby on Rails. Therefore, we are going to configure apache so that it can work with RoR.

INSTALLATION

Let's use curl to download rvm (Ruby Version Manager) to install ruby on your machine and use it. Be careful, the installation is a bit long, and you'll have to enter your root password during the installation.

Important: to use rvm, you must enable the option "run command as a login shell" (edit -> profile preferences), and close/open the terminal.



If everything is okay, enter this command to use the last version of Ruby



Your terminal should ask you to type something like: `source /home/login_x/.rvm/scripts/rvm`. You should write it at the end of your shell configuration file (eg `.bashrc`). Then, source your shell configuration file and **rvm install 2.4**

```
Terminal
~/C-COD-100> rvm use 2.4
```

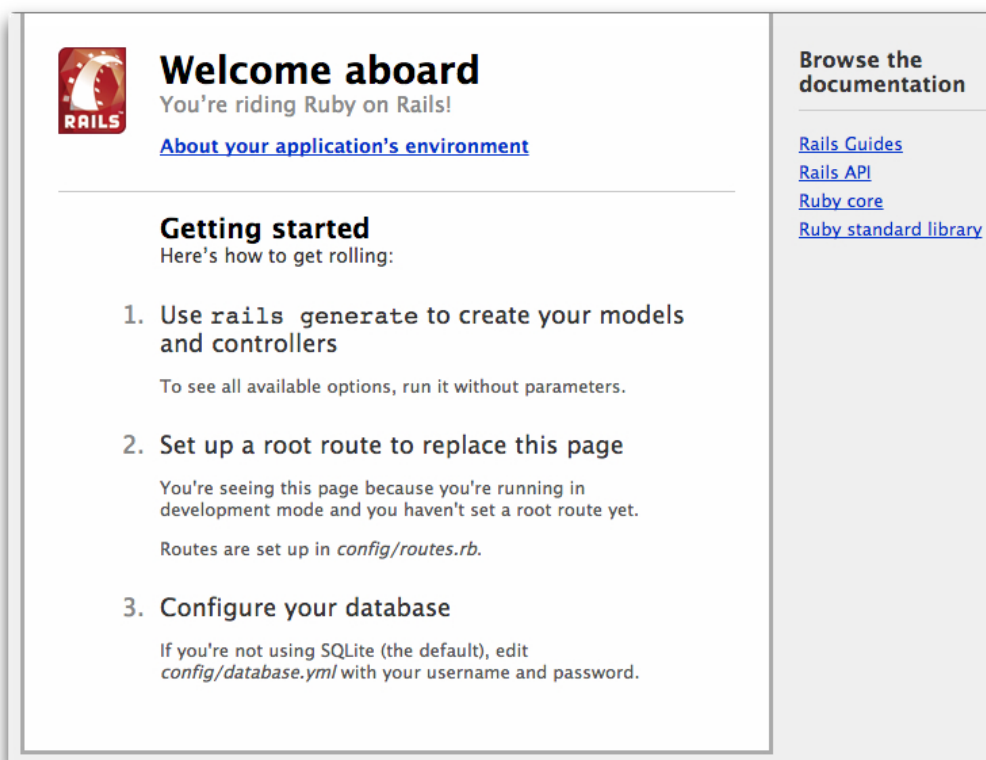
RAILS

We have already installed RVM, Ruby and gem: the package manager for ruby. To install Rails, we are going to use it. All you need to do is:

```
Terminal
~/C-COD-100> rails new superapp
~/C-COD-100> cd superapp
~/C-COD-100> rails s
```

The “**rails**” command starts the server on `http://localhost:3000`, open it on your browser.

You’ll see this page:





+ MAKE & MAKEFILE

The “make” command is a build tool, it is used to ease the compiling process when a large number of files is present and external libraries are used. It reads a Makefile which contains a set of rules and instructions to build a project. It should already be installed on your pc.

TEST

Download the “tree” software sources

```
Terminal
~/C-COD-100> wget http://mama.indstate.edu/users/ice/tree/src/tree-1.7.0.tgz
~/C-COD-100> tar -xvf tree-1.7.0.tgz && cd tree-1.7.0
~/C-COD-100> ls
```

You should see a file named “Makefile”. Now let’s build the software.

```
Terminal
~/C-COD-100> make
gcc -ggdb -DLINUX -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -c -o tree.o tree.c
gcc -ggdb -DLINUX -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -c -o unix.o unix.c
gcc -ggdb -DLINUX -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -c -o html.o html.c
gcc -ggdb -DLINUX -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -c -o xml.o xml.c
gcc -ggdb -DLINUX -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -c -o json.o json.c
gcc -ggdb -DLINUX -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -c -o hash.o hash.c
gcc -ggdb -DLINUX -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64 -c -o colr.o colr.c
gcc -o tree tree.o unix.o html.o xml.o json.o hash.o colr.o
```

Ignore the warnings and test the newly built software.

```
Terminal
~/C-COD-100> ./tree
```

You should see a tree showing files and directories from your current location.



CHEAT SHEET



You will find here the command lines to obtain the different installations. If you do not wish to be spoiled, please do not read the rest of this page.

- Base packages:

```
Terminal
~/C-COD-100> sudo apt-get install -y build-essential libbz2-dev
```

- Curl:

```
Terminal
~/C-COD-100> sudo apt-get install -y curl
```

- Emacs:

```
Terminal
~/C-COD-100> sudo apt-get install -y emacs php-elisp
```

- Vim:

```
Terminal
~/C-COD-100> sudo apt-get install vim
```

- Sublime Text:

```
Terminal
~/C-COD-100> wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg |
sudo apt-key add -
sudo apt-get install apt-transport-https
echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee
etc/apt/sources.list.d/sublime-text.list
sudo apt-get update && sudo apt-get install sublime-text
```

- ssh:

```
Terminal
~/C-COD-100> sudo apt-get install ssh openssh-server
ssh-keygen -t rsa -b 4096
```

- PHP:

```
Terminal
~/C-COD-100> sudo apt-get install php7-common libapache2-mod-php7 php7-cli
php7-curl
```



- MySQL:

```
Terminal
~/C-COD-100> sudo apt-get install mysql-server php5-mysql
```

- NodeJS:

```
Terminal
~/C-COD-100> curl -sL https://deb.nodesource.com/setup_7.x | sudo -E bash -
~/C-COD-100> apt-get install nodejs
```

- RoR:

```
Terminal
~/C-COD-100> curl -sSL https://rvm.io/mpapis.asc | gpg2 -{ -import -
curl -L https://get.rvm.io | bash -s stable -{}-ruby
```

- Python:

```
Terminal
~/C-COD-100> sudo apt-get install python3
```