

Ambiente de Programação com Vagrant

Vagrant na Prática

Instalar Básicos:

Virtualbox e Add ons

<https://www.virtualbox.org/wiki/Downloads>

Vagrant

<https://www.vagrantup.com/downloads.html>

Git

<https://git-scm.com/>

Exportar Box no Linux MINT LMDE 2 (no meu caso)

Baixar comando de

<https://atlas.hashicorp.com/bento/boxes/ubuntu-16.04>

Executar:

```
mkdir vagrant
```

```
cd vagrant
```

```
vagrant init bento/ubuntu-16.04; vagrant up --provider virtualbox
```

Ao terminal executar:

```
vagrant halt
```

```
vagrant package
```

Gerará um package.box e um arquivo Vagrantfile.

Cuidado ao mexer no Vagrantfile, pois o IP de uma box fica gravado no desktop. Mude sempre

Gerará novo backup com esta box configurada

Mudar nome de package.box ubuntu16_1

Adicionar linha de IP fixo no Vagrantfile e o nome da box

E salvar

```
vagrant reload (para carregar as novas configurações)
```

```
vagrant up
```

```
vagrant ssh
```

Esta box está virgem para instalar o php e cia:

```
ribafs@riba ~/ubuntu16_0 $ vagrant reload
```

```
==> default: Clearing any previously set network interfaces...
```

```
==> default: Preparing network interfaces based on configuration...
```

```
default: Adapter 1: nat
```

```
default: Adapter 2: hostonly
==> default: Forwarding ports...
default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Configuring and enabling network interfaces...
==> default: Mounting shared folders...
default: /vagrant => /home/ribafs/ubuntu16_0
==> default: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always will still run.
ribafs@riba ~/ubuntu16_0 $
```

Agora vou instalar os pacotes do php7 e cia e configurações.
Entrarei na vox instalarei. Ao finalarei exportarei novo pacote.

```
vagrant ssh
```

Instalar os pacotes e configurações:

Obs: os pacotes e configurações suportam diversos aplicativos em php, como Joomla, Cakephp, etc

```
sudo su
apt-get update
apt-get install apache2
```

```
# entrar com mysql senha em ambas
apt-get install postgresql
apt-get install mysql-server
```

```
apt-get install libapache2-mod-php7.0
apt-get install php7.0 php7.0-fpm php7.0-mysql
apt-get install mcrypt php7.0-mcrypt php7.0-gd php7.0-pgsql php7.0-ldap
apt-get install php-pear php7.0-xsl curl php7.0-curl phpunit php-xdebug php7.0-intl composer
apt-get install php7.0-zip php7.0-mbstring php-gettext php-mbstring php-auth
apt-get install php-apcu
wget http://ftp.ussg.iu.edu/linux/ubuntu/pool/main/m/memcached/memcached_1.4.25-
2ubuntu1_amd64.deb
dpkg -i memcached_1.4.25-2ubuntu1_amd64.deb
apt-get install php-memcache
```

```
# instalar adminer.org
```

```
php -v
vagrant@vagrant:~$ php -v
PHP 7.0.15-0ubuntu0.16.04.4 (cli) ( NTS )
```

Efetuar novo backup agora com pacotes :

```
exit  
exit  
vagrant halt  
vagrant package
```

Mantenha package.box e Vagrantfile no dir
Copiar para a pasta de backup do vagrant e então renomear a box

Exportar assim e ao final mudar para ub16php7

Criar pasta
/home/ribafs/ub16php7
Contendo
ub16php7.box
Vagrantfile

Executar:

```
cd /home/ribafs/ub16php7  
vagrant reload  
vagrant ssh
```

```
php -v
```

retorna 7.0.15

```
ifconfig
```

```
enp0s8  Link encap:Ethernet  HWaddr 08:00:27:6c:4d:a2  
        inet addr:192.168.0.20  Bcast:192.168.0.255  Mask:255.255.255.0
```

Acessar pelo desktop:

<http://192.168.0.20/adminer.php>

Teoria

O Vagrant é uma incrível e versátil ferramenta que permite a você criar ambientes de desenvolvimento leves, reproduzíveis e portáteis. É fácil de instalar e de configurar.

Se você está usando Windows é uma boa hora para aprender Linux e ter um ambiente de desenvolvimento mais parecido com a maioria dos servidores, que usam Linux, então instale o Vagrant e use ao invés do Xampp, Wamp, etc.

O Vagrant cria uma máquina virtual, que ele chama de box, com uma distribuição Linux, onde você pode instalar tudo que quiser para que seu ambiente fique redondinho.

O Vagrant é indicado para:

- quem deseja aprender sobre ferramentas linux;
- testar ambientes de teste antes de colocar em produção
- levantar um servidor LAMP em apenas 30 minutos
- garantir trabalho em equipe sem diferentes versões de softwares, nem diferença de servidores

As box são ambientes leves, que gastam bem pouco recurso do seu desktop. Com 1GB de RAM você cria um bom ambiente, pois não precisam nem ter uma interface gráfica, apenas a console acessada via ssh.

Caso cometa algum erro e perca o controle sobre alguma porta ou serviço do Windows algumas vezes precisa formatar seu desktop e instalar o Windows novamente.

Caso tenha algum problema com sua box, você a destrói e reconstrói novamente com facilidade.

Vantagens de uma box do Vagrant em relação a uma VM do Virtualbox:

- Muito rápida de usar
- Simulam o servidor
- São bem leves, pois uma única CPU e 1GB de RAM são mais que suficientes
- São fáceis de transportar:

```
vagrant halt
```

```
vagrant package --output nomedabox.box
```

Vantagens do Vagrant:

- Fácil de criar várias box, cada uma com uma versão do PHP, uma com o Ubuntu 14.04 (5.5.9) e outra com o 16.04 (7.0.8)
- Fácil de testar várias distribuições ou versões, servidores web (Apache ou Nginx), SGBDs (MySQL ou PostgreSQL)
- São facilmente destrutíveis. Caso tenha algum problema destrua e crie outra:

```
vagrant halt
```

```
vagrant destroy
```

vagrant ...

- Teste o desempenho do seu código em vários ambientes e SO.
- Compartilhar seu ambiente com outros integrantes da equipe
- Configurar sua box de forma semelhante ao servidor para evitar conflitos
- É muito parecido com o servidor
- Ensina como usar o servidor pelo terminal e ferramentas como o SSH
- Você usa a distribuição que quiser, o SGBD que quiser e o servidor web que desejar
- Instale o Vagrant em qualquer sistema operacional
- Podemos baixar algumas box que já vem com muita coisa pronta, como é o caso do Homestead

para o laravel ou friendsofcake

Criando Box com CentOS

Solução para erro ao montar pasta compartilhada:

Após atualizar o CentOS da box execute:

```
vagrant up  
vagrant ssh  
sudo yum -y install kernel-devel  
exit  
vagrant provision --provision-with additions  
vagrant reload
```

vagrant ssh

Exportando uma box

Após tudo instalado, configurado e testado na box é hora de exportar a box para guardar ou para transportar para outro computador, de casa, do trabalho, de um curso onde irá ministrar, de uma palestra, etc.

- Acesse a box
- vagrant halt
- vagrant package --output nomedabox.org

Ferramentas para Vagrant com Windows

O windows não tem cliente de SSH, portanto precisamos de uma outra ferramenta que contenha.

O Git contém

O cmdr também.

<http://cmdr.net/>

Basta o mini.

Configurando o Windows

Painel de Controle - Sistema e Segurança - Alterar configurações do UAC (User Access Control)

Mover slider para Nunca notificar

Reiniciar computador

Algumas boxes:

FriendsOfCake (CakePHP - 2 anos atrás)

```
vagrant init friendsofcake/cakephp-baking; vagrant up --provider virtualbox
```

Lorenzo - 2 anos

```
vagrant init lorenzo/php-baking; vagrant up --provider virtualbox
```

Ironbone - 1 mês - PHP7

```
vagrant init ironbone/cakephp; vagrant up --provider virtualbox
```

Chiemi - 1 ano - Centos 7

```
vagrant init chiemi627/centos7.0-cakephp-mysql; vagrant up --provider virtualbox
```

Cakebox - 1 ano - <https://github.com/alt3/cakebox>

```
vagrant init alt3/cakebox; vagrant up --provider virtualbox
```

Ubucake (cake3 e outros recursos)

```
vagrant init tanuck/ubucake; vagrant up --provider virtualbox
```

Ctisvs - 4 meses

```
vagrant init ctisvs/cakephp; vagrant up --provider virtualbox
```

Tlamin - 13 dias atrás: joomla, wordpress e drupal

```
vagrant init tlamin/it610; vagrant up --provider virtualbox
```

Ives - 1 ano atrás: joomla com virtuemart

```
vagrant init ives/joomlavirtualmart; vagrant up --provider virtualbox
```

Homestead (Laravel) - 3 meses atrás

```
vagrant init laravel/homestead; vagrant up --provider virtualbox
```

Existem ferramentas que colaboram com a configuração das box para o Vagrant.:

Puppet, Chef, Ansible e PuPHPet (este último gera código compatível com o Windows, os demais não).

Plugin do vagrant que cria automaticamente o hostname quando a box é startada:

```
vagrant plugin install vagrant-hostsupdater
```

Virtualbox Guest

```
vagrant plugin install vagrant-vbguest
```

Corrigindo instalação de plugin

```
sudo rm -rf ~/.vagrant.d
```

```
vagrant plugin update
```

```
vagrant up --provision
```

Para criar hostname para cada box criada:

No linux

```
sudo nano /etc/hosts
```

```
192.168.0.10 cake3.dev cake3
```

No Windows

```
notepad c:\windows\system32\drivers\etc\hosts
```

```
192.168.0.10 cake3.dev cake3
```

Baixando box do Vagrant

<https://atlas.hashicorp.com/boxes/search>

Aqui você escolhe o perfil da box que deseja instalar e pega o comando e executa num diretório pelo terminal. Exemplo:

Exemplo:

<https://atlas.hashicorp.com/bento/boxes/centos-7.2>

```
mkdir centos72
```

```
cd centos72
```

```
vagrant init bento/centos-7.2; vagrant up --provider virtualbox
```

Após instalar com sucesso

`vagrant ssh` (para acessar a box criada)

Do PuPHPet

<https://puphpet.com/>

Neste você escolhe a distribuição e os pacotes a instalar e ao final ele gera o pacote zip para a instalação local da box.

Criando uma box:

Abrir terminal

```
mkdir vagrant
```

```
cd vagrant
```

```
vagrant init (ele criará um arquivo padrão Vagrantfile no diretório atual)
```

É uma boa hora de editar o Vagrantfile, descomentar alguns trechos (shared folder, network private, forward de portas, hostname, etc)

Acessar o

<https://atlas.hashicorp.com/boxes/search> para escolher uma box

Remova o Vagrantfile criado e para instalar o Ubuntu 14.04 execute:

```
vagrant init bento/ubuntu-14.04
```

```
vagrant up --provider virtualbox
```

Após instalar:

```
vagrant ssh
```

Verificando as configurações de SSH

```
vagrant up
```

`vagrant ssh-config`

Listar boxes

`vagrant box list`

Removendo uma das box listadas

`vagrant box remove nome.box`

Comandos

`vagrant up` - iniciar a máquina virtual e executa os provisioners definidos, partindo das definições do Vagrantfile

`vagrant ssh` - acessar a máquina virtual via ssh. Não requer login nem senha

`vagrant provision` - roda somente o provisioner, sem reiniciar a vm. Útil após alterações no Vagrantfile ou provisioner.

`vagrant reload` - reinicia a vm e atualiza alterações no Vagrantfile

`vagrant destroy` - exclui a máquina virtual

`vagrant halt` - desliga a vm.

`vagrant init` - cria um novo Vagrantfile, caso não exista um.

`vagrant status`

`vagrant box list` – lista as boxes criadas

`vagrant -v` – mostra versão do vagrant

`vagrant up --provider virtualbox` – reiniciar e carregar as mudanças do Vagrantfile

Ver todos os comandos:

`vagrant`

`ribafs@ribalinux:~/projetos$ vagrant`

Usage: `vagrant [options] <command> [<args>]`

`-v, --version` Print the version and exit.

`-h, --help` Print this help.

Common commands:

<code>box</code>	manages boxes: installation, removal, etc.
<code>connect</code>	connect to a remotely shared Vagrant environment
<code>destroy</code>	stops and deletes all traces of the vagrant machine
<code>global-status</code>	outputs status Vagrant environments for this user
<code>halt</code>	stops the vagrant machine
<code>help</code>	shows the help for a subcommand
<code>hostsupdater</code>	
<code>init</code>	initializes a new Vagrant environment by creating a Vagrantfile
<code>login</code>	log in to HashiCorp's Atlas
<code>package</code>	packages a running vagrant environment into a box
<code>plugin</code>	manages plugins: install, uninstall, update, etc.
<code>port</code>	displays information about guest port mappings
<code>powershell</code>	connects to machine via powershell remoting
<code>provision</code>	provisions the vagrant machine
<code>push</code>	deploys code in this environment to a configured destination
<code>rdp</code>	connects to machine via RDP

reload	restarts vagrant machine, loads new Vagrantfile configuration
resume	resume a suspended vagrant machine
share	share your Vagrant environment with anyone in the world
snapshot	manages snapshots: saving, restoring, etc.
ssh	connects to machine via SSH
ssh-config	outputs OpenSSH valid configuration to connect to the machine
status	outputs status of the vagrant machine
suspend	suspends the machine
up	starts and provisions the vagrant environment
vbguest	
version	prints current and latest Vagrant version

For help on any individual command run ``vagrant COMMAND -h``

Additional subcommands are available, but are either more advanced or not commonly used. To see all subcommands, run the command ``vagrant list-commands``.

Definindo a box no Vagrantfile

```
Vagrant.configure(2) do |config|
  # https://docs.vagrantup.com.

  # boxes at https://atlas.hashicorp.com/search.
  #config.vm.box = "/backup/dnocs6_bento.box"
  config.vm.box = "D:/Tools/Vagrant/Backup/dnocsdev-1.1.box"

  config.vm.network "forwarded_port", guest: 80, host: 8080
  config.vm.network "private_network", ip: "192.168.0.4"
  #config.vm.synced_folder "/backup/web", "/var/www/html"
  config.vm.synced_folder "c:/xampp/hdtocs/web", "/var/www/html"

  config.vm.provider "virtualbox" do |vb|
    vb.memory = "1024"
  end

  #config.vm.provider "virtualbox" do |vb|
  #  # Display the VirtualBox GUI when booting the machine
  #  vb.gui = true
  #
  #  # Customize the amount of memory on the VM:
  #  vb.memory = "1024"
  #end
end
```

Caso tenhamos uma box com modo gráfico devemos descomentar:

```
config.vm.provider "virtualbox" do |vb|
  vb.gui = true
end
```

Now create a new folder somewhere (for your new project that uses the box) and initialize a new default Vagrantfile via

```
vagrant init
```

Open the file, and edit this: Rename the default boxname (which might be **precise64**) to something new, like “**my-zend-framework-box**”.

```
config.vm.box = "my-zend-framework-box"
```

Uncomment and edit the default box location from

```
# config.vm.box_url = "http://domain.com/path/to/above.box"
```

to the filepath of your package.box. Windows users need to write the path this way:

Advertisement

```
config.vm.box_url = "file:///d:/folder/package.box"
```

Then uncomment and change the default IP to something you want, I recommend increasing the last number of the IP.

```
config.vm.network :private_network, ip: "192.168.33.101"
```

Run your Vagrant box with:

```
vagrant up
```

Create a new VM from the new box file (long method)

If you want to add the box to your vagrant box list (to use the box by it's name, not by giving the file location) do it like this:

```
vagrant package
```

Add the box to Virtualbox (chose a box name for **name-of-this-box**):

```
vagrant box add name-of-this-box package.box virtualbox
```

Now you can create virtual machines from this box by simply giving the name of the box in the Vagrantfile, like

```
config.vm.box = "name-of-my-box"
```

A **config.vm.box_url** is not necessary anymore.

Acessando a box do desktop e vice-versa

Adicione a box no hosts do desktop

Linux - /etc/hosts

Para salvar o hosts do windows precisa parar o antivírus e executar:
notepad c:\windows\system32\drivers\etc\hosts

```
192.168.0.4 dnocs.dev www.dnocs.dev
```

Exemplo: minha máquina tem o IP – 10.0.0.128.

Então posso acessar os recursos da box assim:

Acessar a box do desktop via SSH

```
cd /home/ribafs/projetos
```

```
vagrant ssh
```

Acessar a box via Web

http://192.168.0.4
http://localhost:8080
http://dnocs.dev

Acessar os arquivos da box pelo Nautilus ou pelo Windows Explorer

Meu	Box
/backup/web	/var/www/html
/home/ribafs/projetos	/vagrant

O que criar aqui aparece lá e vice-versa.

Efetuar o download de box do atlas

Acessar
<https://atlas.hashicorp.com/bento/boxes/ubuntu-16.04>

Perfil - bento
Box - ubuntu-16.04
Provider - virtualbox
Anotar a versão da box - 2.2.7

Resulta em:
<https://atlas.hashicorp.com/bento/boxes/ubuntu-16.04/versions/2.2.7/providers/virtualbox.box>

Efetuar o download de package.box para /home/ribafs

Adicionar para o Vagrantfile:
vagrant box add bento/ubuntu-16.04 /home/ribafs/package.box

vagrant init package

Editar o Vagrantfile e efetuar os ajustes

vagrant up

Limpar cache do Vagrant
Verificar
vagrant global-status
Limpar as abandonadas
vagrant global-status --prune

Sintaxe do synced_folder

O segundo parâmetro é um diretório na máquina virtual (guest) e com path absoluto.
O primeiro parâmetro (web/) é um diretório na máquina desktop (host) e é um path relativo ao raiz do projeto.
A documentação diz path relativo mas uso também path absoluto e funciona
/var/www/html => /backup/web

