VAGRANT

devops, info

- vagrantfile
- vagrantfile examples

related

- glusterfs vagrant POC vagrant, vagrantfile
- vagrant (openwebinars) devops, cursos, vagrant

help

- https://www.vagrantup.com/docs/vagrantfile/machine_settings
- vagrant_hosts → `sync_hosts`: https://www.rubydoc.info/gems/vagrant-hosts/2.6.2

config files

- config.vm : https://www.vagrantup.com/docs/vagrantfile/machine_settings.html
- provision docker : https://www.vagrantup.com/docs/provisioning/docker.html

resumen comandos

imágenes (box)

- `vagrant box add <imagen>`: descargar
  - alternativamente, descarga manual: `wget https://app.vagrantup.com/<USER>/boxes/<BOX>/versions/<VERSION>/providers/<PROVEEDOR>.box`
- `vagrant box list`: listar imagenes disponibles
- `vagrant box repackage <imagen> <proveedor> <version>`: generar fichero .box a partir de imagen descargada
- `vagrant box remove <imagen>`: eliminar imagen
- `vagrant box add <fichero.box> –name debian/jessie64`: añadir imagen desde fichero local (nombre ejemplo)
- `vagrant box outdated`: mirar estado de versión de las imágenes descargadas
  - `–global` sobre todas las imágenes
- `vagrant box prune`: eliminar versiones de imágenes viejas (cuando se actualiza no se elimina la versión anterior)
- `vagrant box update`: actualizar imágenes
- crear una imagen desde una VM de virtualbox:
  - `VBoxManage list vms`: obtener listado de MVs
  - `vagrant package –base “Nombre de la MV” –output maquina.box`

máquinas virtuales

- `vagrant init <imagen>`: montar un escenario (en su directorio)
  - -m crea el `Vagrantfile` sin comentarios

- `vagrant up`: arrancar la máquina
- `vagrant reload`: aplica cambios en el Vagrantfile sin que impliquen la destrucción de la máquina.
- `vagrant halt`: parar la máquina
- `vagrant status`: estado de la máquina
- `vagrant global-status`: estado y situación de todas las máquinas
- `vagrant ssh`: conexión a la máquina
- `vagrant destroy`: destruir máquina

red

- `vagrant port`: muestra los puertos redireccionados de la MV

snapshots

- `vagrant snapshot save <nombre_snapshot>`: crea un snapshot
- `vagrant snapshot restore <snapshot>`: restaura un snapshot
- `vagrant snapshot push` ↔ `vagrant snapshot pop`: crea una «pila» de snapshots

Vagrantfile

- `vagrant validate` : comprueba que el fichero **Vagrantfile** sea correcto

otros

- `vagrant rsync`: sincroniza los directorios «compartidos» entre el host y la VM
- `vagrant rsync-auto` : demonio que se encarga de mantener los directorios «compartidos» en syncro contínua

plugins

vagrant-hosts

- `vagrant plugin install vagrant-hosts` → https://www.rubydoc.info/gems/vagrant-hosts/2.6.2

```
Vagrant.configure('2') do |config|
  config.vm.box = "puppetlabs/ubuntu-14.04-64-nocm"

  config.vm.provision :hosts do |provisioner|
    # Add a single hostname
    provisioner.add_host '10.0.2.2', ['myhost.vagrantup.internal']

    # Or a fqdn and a short hostname
    provisioner.add_host '10.0.2.3', ['myotherhost.vagrantup.internal',
'myotherhost']

    # Or as many aliases as you like!
    provisioner.add_host '10.0.2.5', [
      'mypuppetmaster.vagrantup.internal',
      'puppet.vagrantup.internal',
      'mypuppetmaster',
      'puppet'
```

```
    ]
  end
end
```

disk-size

- `vagrant plugin install vagrant-disksize` : ayuda a crear discos «a medida»

```
Vagrant.configure("2") do |config|
    config.vm.box = "debian/stretch64"
    config.disksize.size = "20GB"
end
```

  - https://unix.stackexchange.com/questions/176687/set-storage-size-on-creation-of-vm-virtualbox

uso en vim

```
" Teach vim to syntax highlight Vagrantfile as ruby
"
" Install: $HOME/.vim/plugin/vagrant.vim
" Author: Brandon Philips <brandon@ifup.org>

augroup vagrant
  au!
  au BufRead,BufNewFile Vagrantfile set filetype=ruby
augroup END
```

vagrant + virtualbox

como enlazar una máquina Vagrant con su correspondiente máquina en el proveedor (suponiendo que sea virtualbox)

- `vagrant global-status` : relación de máquinas y su path **original**
- `VBoxManage list vmw` : ralación de máquinas virtuales y su ID
- `.vagrant/machines/default/virtualbox/action_provision` : contiene el ID de la máquina virtualbox asociada

solo hay que localizar esos ficheros, extraer el campo del ID y cotejarlo con la información de **VBoxManage**:

```
find /home/mate/Vagrant -name action_provision
```

```
for var in $(find ~/Vagrant -name action_provision); do echo "${var}"; cat $var |
cut -d ":" -f 2 | sort -n; done
```

```
VBoxManage list vms | sed 's/[{}"]//g' | awk '{print $2 " " $1}'
```

No funciona correctamente

```
LISTA=`vagrant global-status | awk '/^[a-f0-9]{1,7}\s+/ {print $5}'`
```

```
for variable in $LISTA; do cat
${variable}/.vagrant/machines/default/virtualbox/action_provision; done
```

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/tech:vagrant:start

Last update: **14/02/2022 18:48**