

Vagrant and Ansible

Esercitazione

The Open Source Tools selected

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

- A free software tool for creating customizable, lightweight, reproducible, and portable development environments made up of Virtual Machine Images (VMIs).
 - *(Windows, Linux, Macintosh)*



■ Ansible <http://docs.ansible.com>

- An IT automation tool to automatically keep different development environments aligned - in our case the ones of the MIDAS project partners.
 - *(installed inside the Guest Virtual Machine)*



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

- A powerful x86 and AMD64/Intel64 virtualization product that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2.
 - *(Windows, Linux, Macintosh)*





....more on Vagrant

■ Getting started with Vagrant

- Open a terminal or command prompt and type:
 - *\$ vagrant -v*
- To initialize a VM configuration starting by an existing Virtual Machine Image, you can type:
 - *\$ vagrant init precise64 <http://files.vagrantup.com/precise64.box>*
- To running your VM :
 - *\$ vagrant up*
- Ok, now a Ubuntu 12.04 LTS 64-bit is running in your local machine! If you want access to it, you can type:
 - *\$ vagrant ssh*



....more on Vagrant

■ Suspend, Halt, Destroy your local Virtual Machine

■ Suspending the local VM

- `$ vagrant suspend`

Save the current running state of the VM and stop it. You can resume your VM by typing
`$ vagrant up`

Pro: only 5 or 10 seconds to stop and start your work
Cons: disk space to store the VM and its status

■ Halting the local VM

- `$ vagrant halt`

Gracefully shutdown the guest OS. You can resume your VM by typing
`$ vagrant up`

Pro: no disk space to store the status of VM
Cons: extra time to start from a cold boot

■ Destroying the local VM

- `$ vagrant destroy`

Remove all traces of the guest machine from your system. You can reconfigure your VM by typing
`$ vagrant up`

Pro: the disk space is left clean
Cons: extra time to re-import and re-provision the VM



Vagrant + Ansible

- **To use Ansible with Vagrant you can type**

- `$ vagrant up`

The Ansible playbook is automatically installed (the first time) and run during calls this command. The VM configuration expressed in the YAML format is read and executed.

- `$ vagrant provision`

This command allows you to re-configure an already running virtual machine to just run the provisioner. This allows, for example, to run apt-updates on the running VM, or to update the packages behind the running VM



Vagrant + Ansible

■ VM Hadoop

```
hadoop_vm
├─ Vagrantfile
├─ bootstrap.sh
├─ local_vars.rb
├─ provisioning
│   └─ files
│       ├── core-site.xml
│       ├── hadoop-env.sh
│       ├── hdfs-site.xml
│       ├── id_dsa
│       ├── id_dsa.pub
│       └─ mapred-site.xml
├─ playbook.yml
└─ repository
    └─ hadoop-1.2.1.tar.gz
```



Vagrant + Ansible

- **Ansible doc & modules**
- **Apt module http://docs.ansible.com/apt_module.html**
- **Template module http://docs.ansible.com/template_module.html**
- **Command module http://docs.ansible.com/command_module.html**
- **Lineinfile module http://docs.ansible.com/lineinfile_module.html**
- **Unarchive module http://docs.ansible.com/unarchive_module.html**
- **Authorized_key module**
http://docs.ansible.com/authorized_key_module.html

LET'S START

Alberto De Francesco

alberto.defrancesco@isti.cnr.it