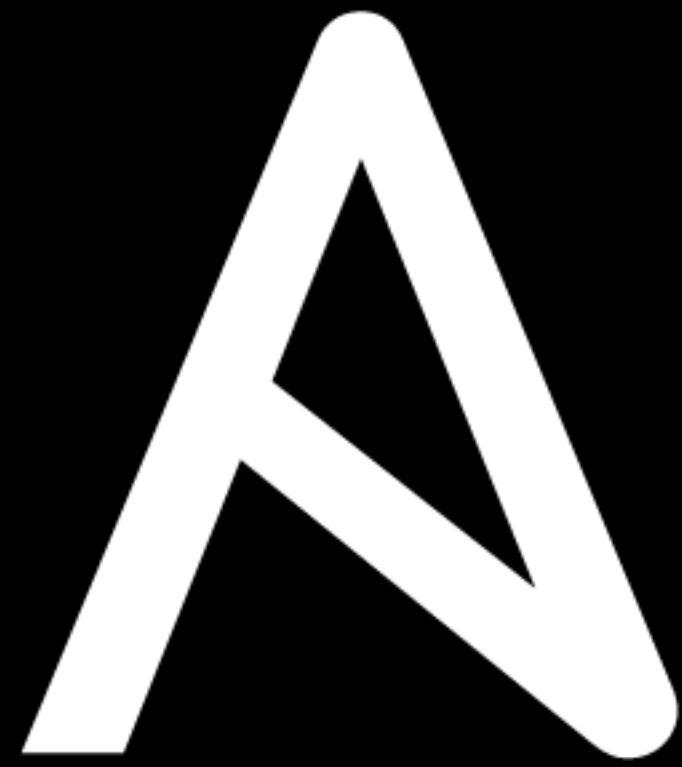


Quickstart

```
git clone https://github.com/bbaassssiiee/vagransible
```

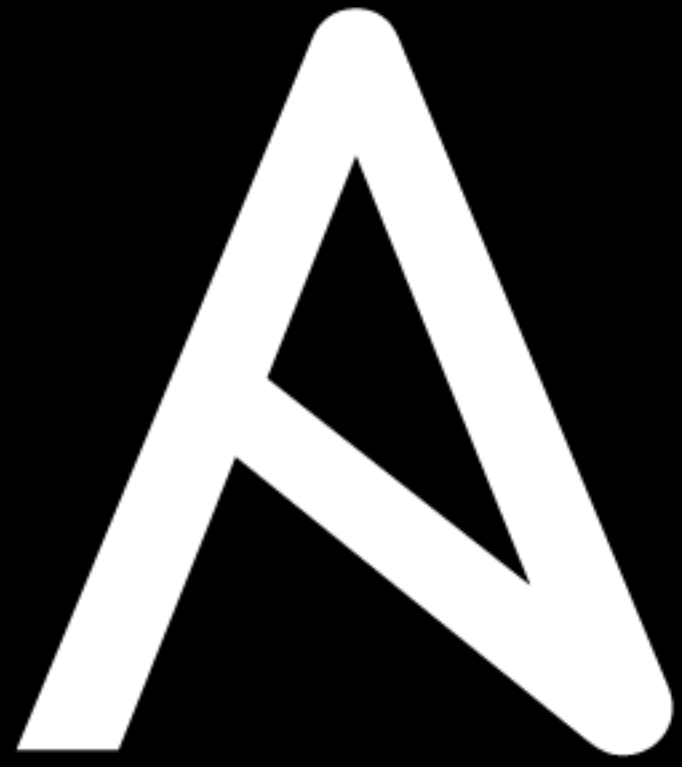
```
cd vagransible
```

```
vagrant up centos6
```



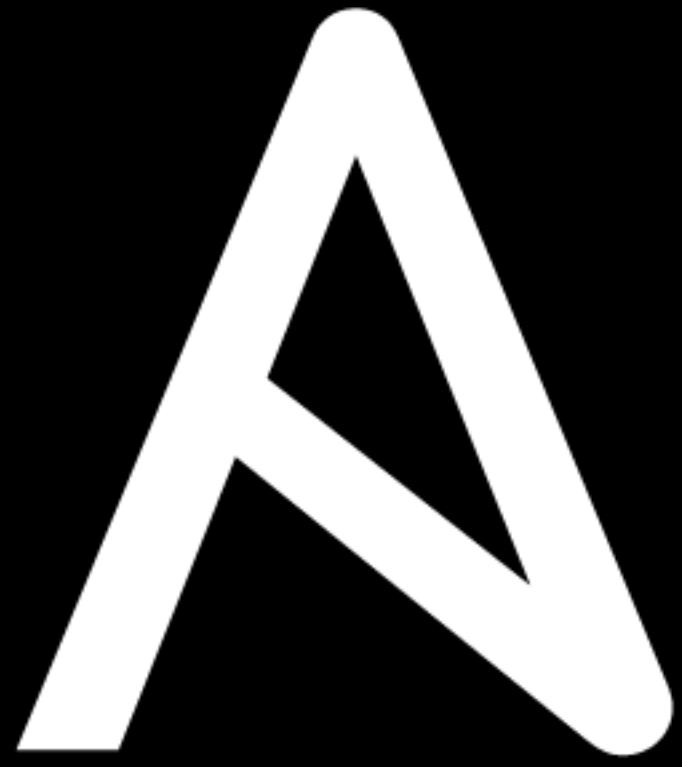
Ansible + Vagrant

- Vagrant
- Providers
- Provisioners
- Configuration as data
- Examples



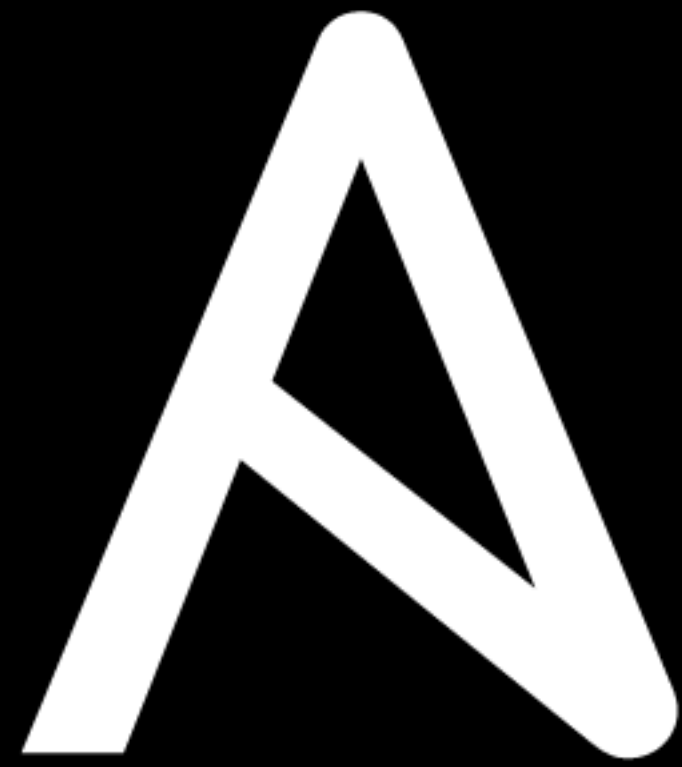
Vagrant

- Reproducible environments
- Cross-platform
- Virtual Machines & Docker
- Configuration as code
- Multilevel access
- Simple & Powerful



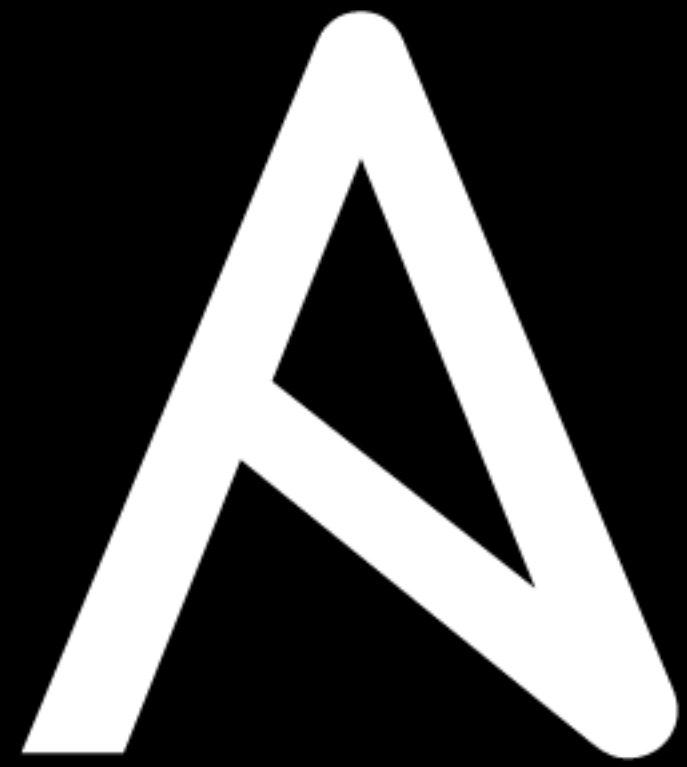
What do you need?

- 8Gb Ram or more, SSD, Vagrant
- VMWare Fusion, VirtualBox,
- Python, SSH client, git client



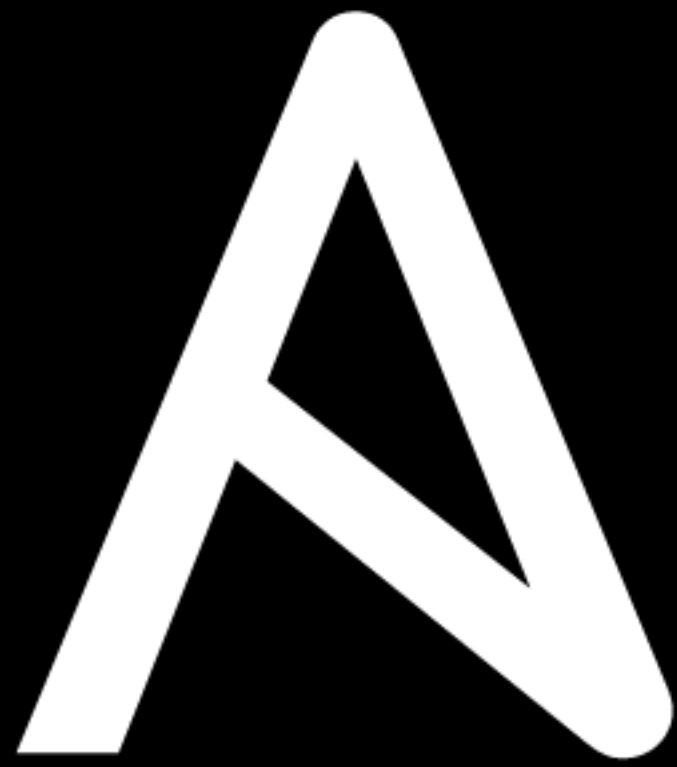
Provider agnostic

- VirtualBox
- VMWare
- Amazon Web Service
- Docker
- Microsoft Hyper-V
- ...



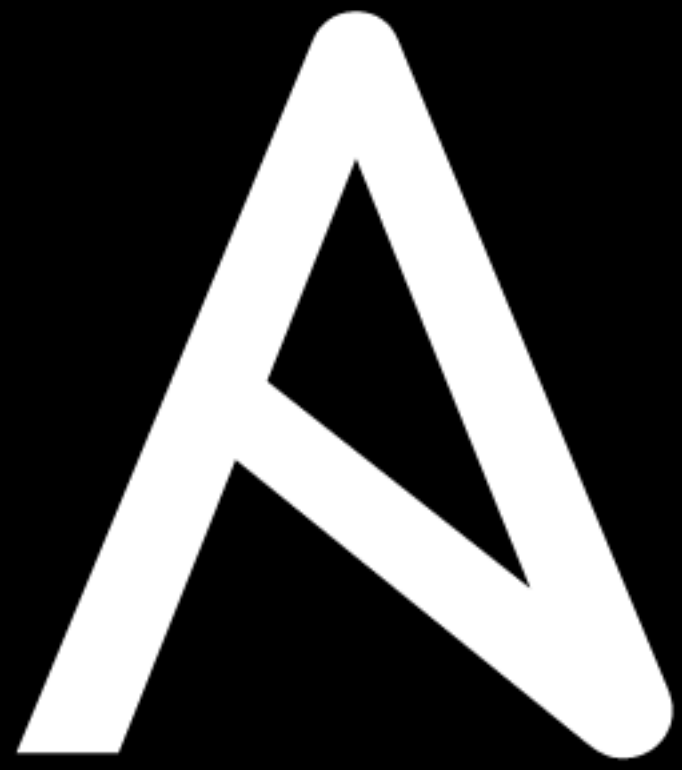
Provisioner agnostic

- Ansible
- Salt
- Puppet
- Chef
- bash
- ...



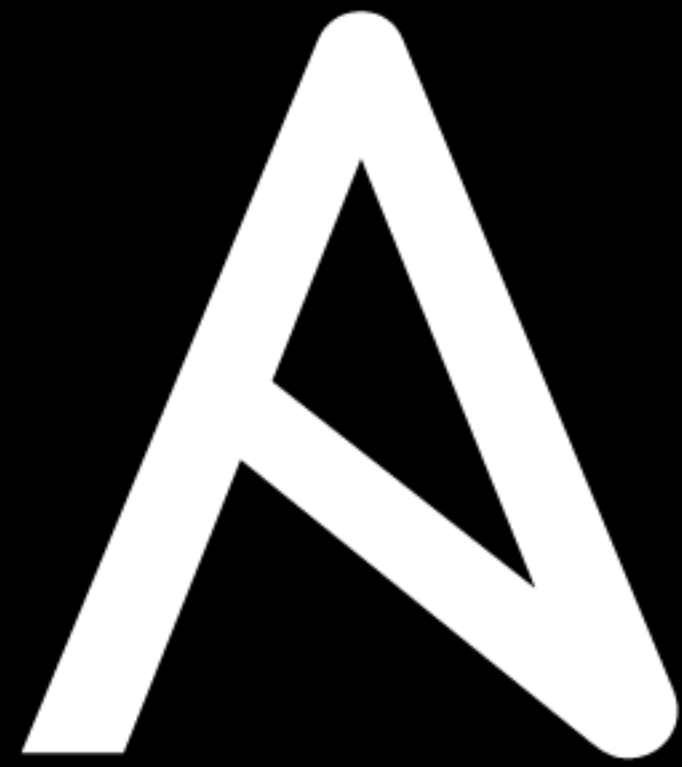
vagrant up

```
environment
kreta:environment bas$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'chef/centos-6.5'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'chef/centos-6.5' is up to date...
==> default: Setting the name of the VM: PRODUCT-123
==> default: Fixed port collision for 22 => 2222. Now on port 2201.
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 => 2201 (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2201
    default: SSH username: vagrant
    default: SSH auth method: private key
    default: Warning: Connection timeout. Retrying...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
==> default: Mounting shared folders...
    default: /vagrant => /Users/bas/code/environment
==> default: Running provisioner: ansible...
```

vagrant <commands>

- **vagrant up** - starts the machine, possibly downloading and caching the box image & provisioning the VM
- **vagrant ssh** - logs you into the VM
- **vagrant halt** - stops the VM
- **vagrant suspend** - pauses the VM
- **vagrant destroy** - trashes the VM

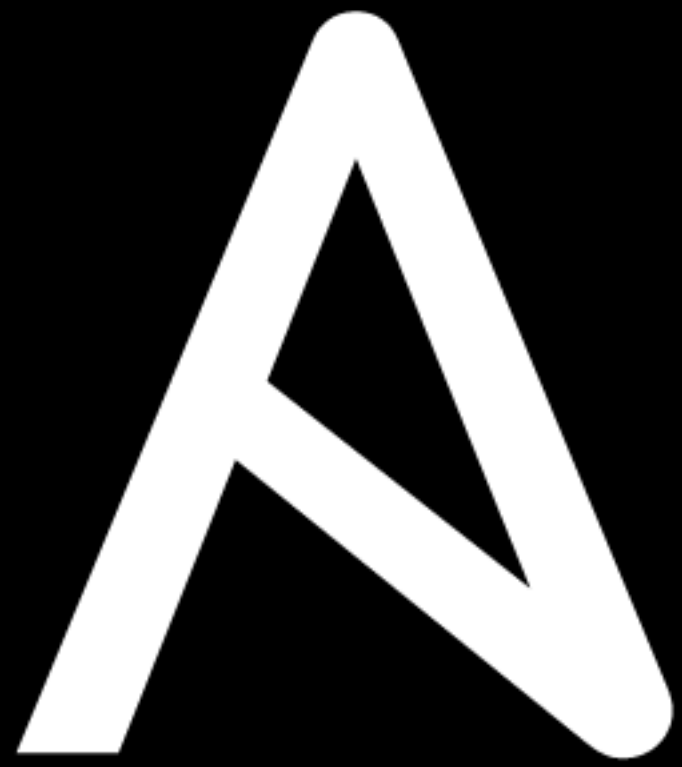


Vagrantfile

```
VAGRANTFILE_API_VERSION = "2"
```

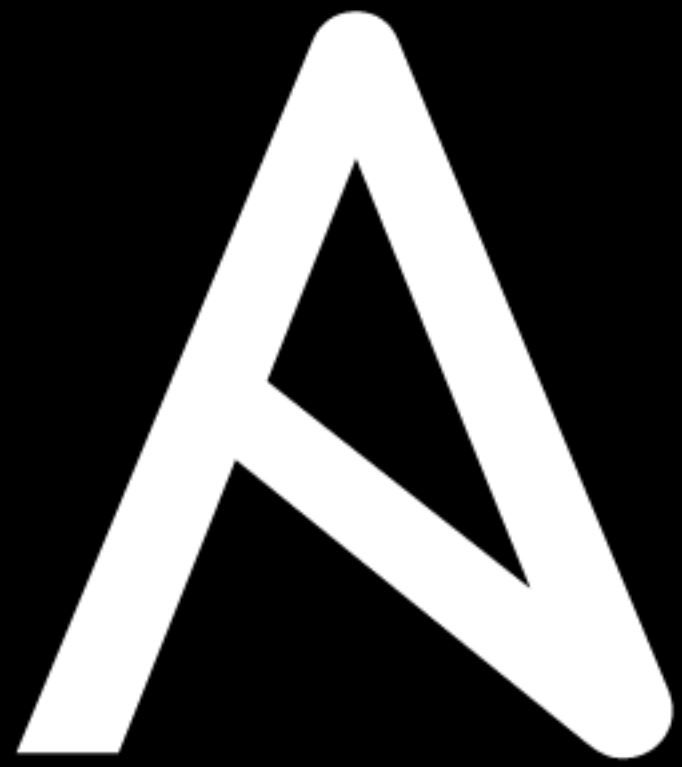
```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |  
  config|
```

```
end
```



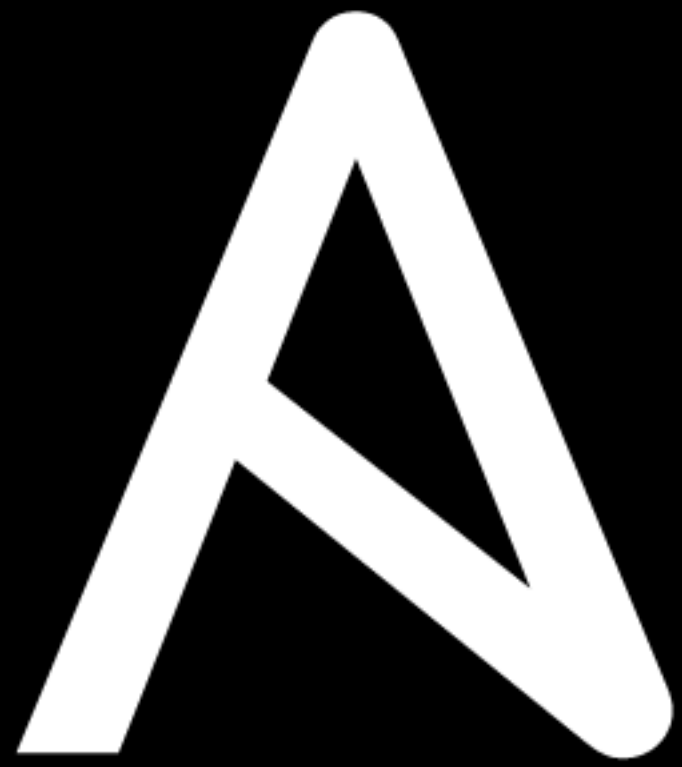
provision ansible

```
config.vm.provision "ansible" do |ansible|  
  ansible.inventory_path = "ansible.ini"  
  ansible.playbook = "provision.yml"  
  
  ansible.verbose = "vv"  
  
end
```



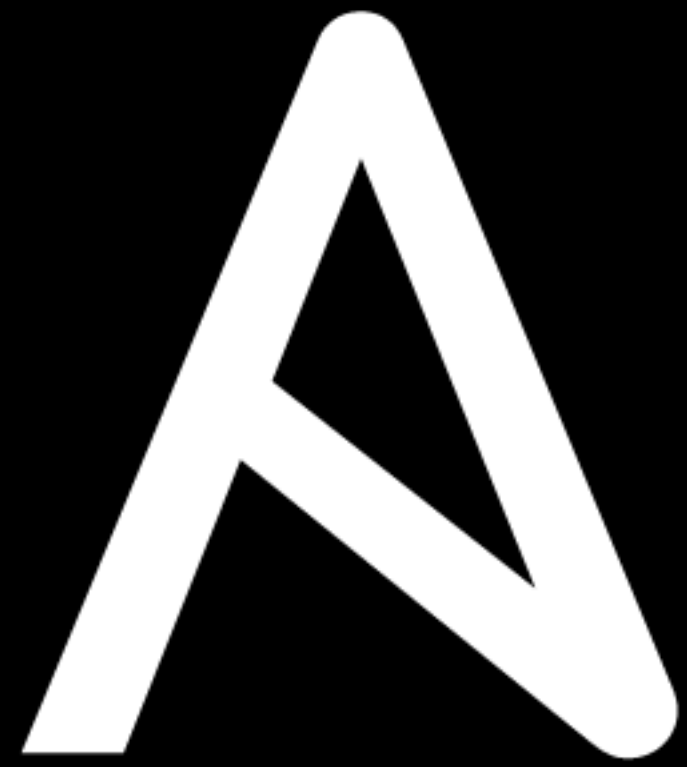
provider virtualbox

```
config.vm.provider "virtualbox" do |vb|  
  vb.gui = false  
  vb.customize ["modifyvm", :id, "--memory", 2048]  
  
  vb.name = "centos6"  
  
end
```



provider vmware_fusion

```
config.vm.provider "vmware_fusion" do |vmware|  
  vmware.gui = false  
  
  vmware.vmx["memsize"] = "2024"  
  
  vmware.vmx["numvcpus"] = "2"  
  
end
```

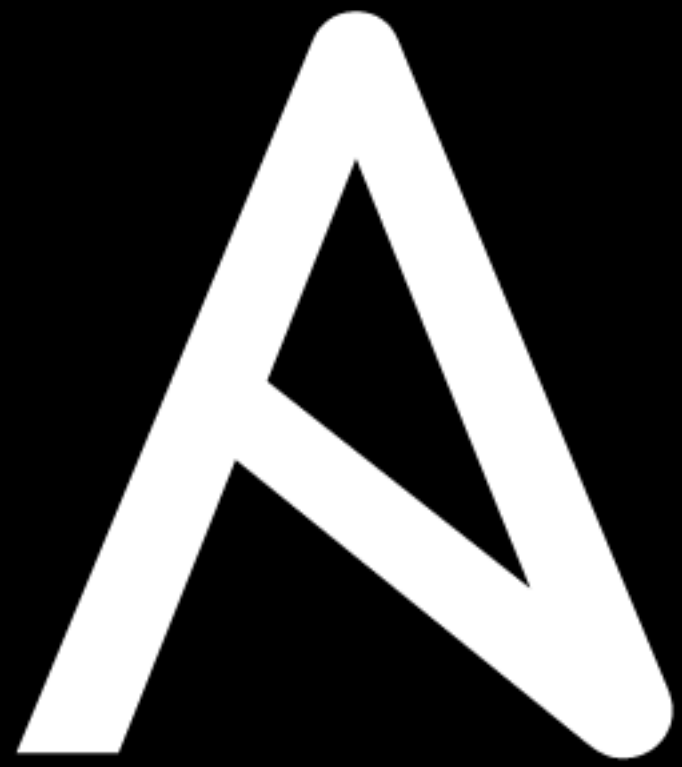


provider selection

```
# Prefer VirtualBox before VMware Fusion
```

```
config.vm.provider "virtualbox"
```

```
config.vm.provider "vmware_fusion"
```



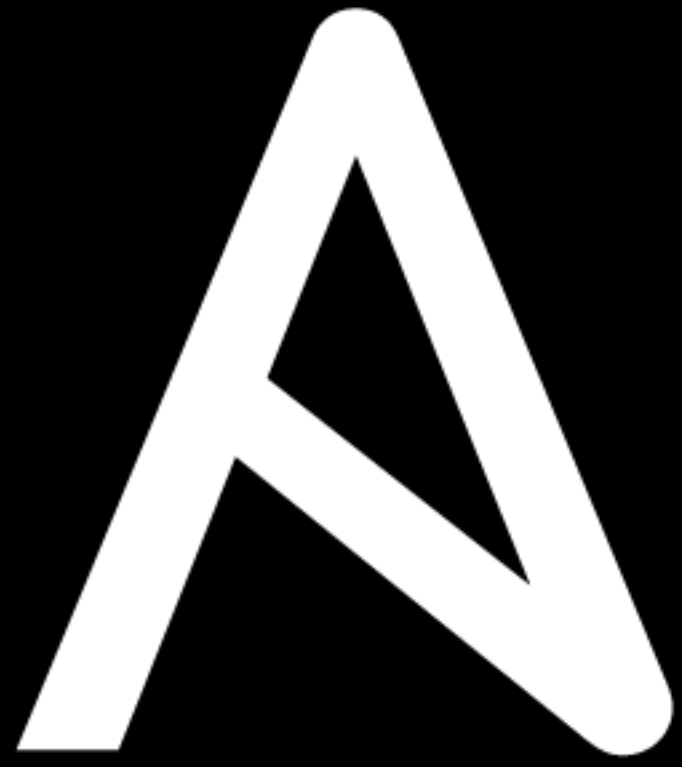
config options

```
config.ssh.insert_key = false
```

```
config.vm.box_check_update = false
```

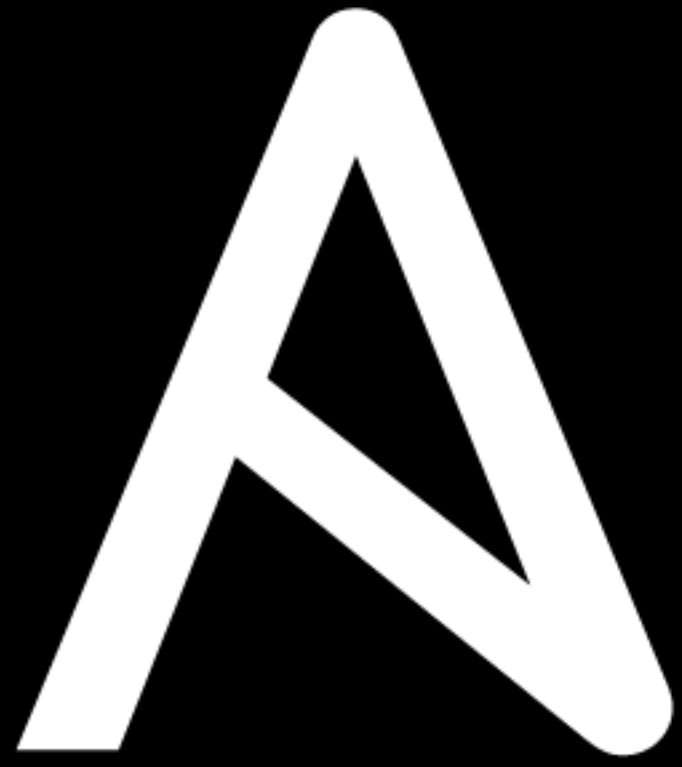
```
#disable guest additions
```

```
config.vm.synced_folder ".", "/vagrant", id:  
"vagrant-root", disabled: true
```



Ansible

- Easiest IT automation to use, ever.
- Minimal learning curve
- Easy audit/review/rewrite of content
- Minimal requirements: SSH & python 2.5+
- No daemons, no master, no SPOF
- Secure, fast, scalable
- Pluggable, Extensible



Resources

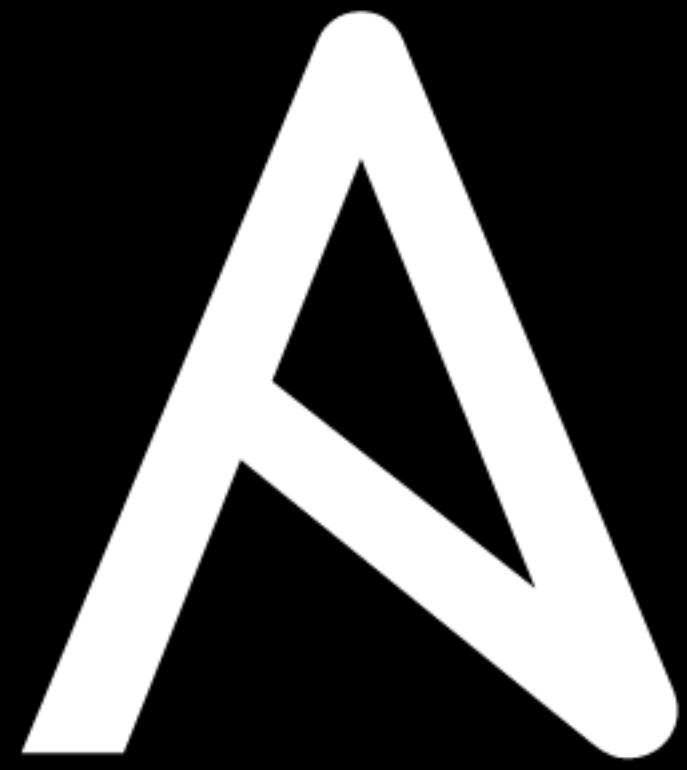
@bbaassssiiee

<https://github.com/bbaassssiiee>

<http://www.meetup.com/Ansible-Benelux>

<https://galaxy.ansible.com>

<http://www.vagrantup.com>



ANSIBLE

Benelux Meetup

Bas Meijer @bbaasssiiee 2015-02-28 Utrecht