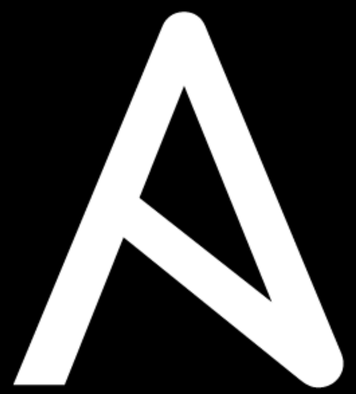


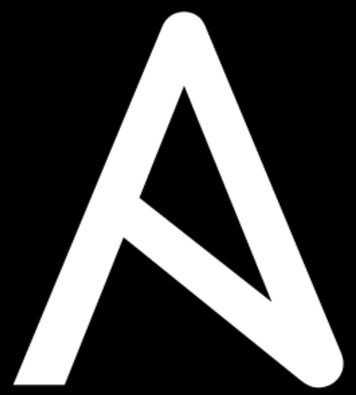
ANSIBLE

Benelux Meetup

Bas Meijer @bbaassssiiee 2015-05-27 Amsterdam

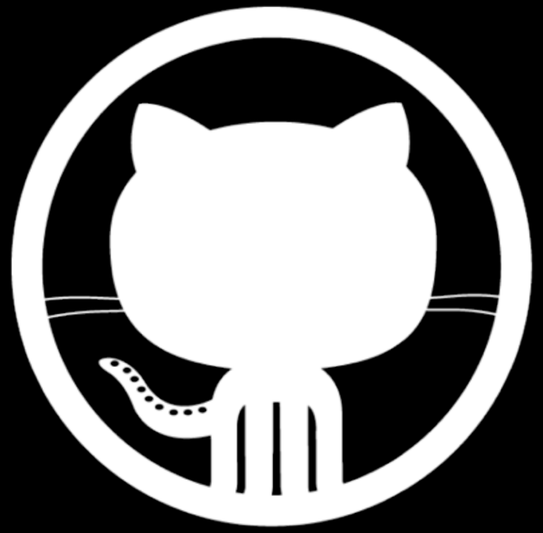
Fake IT, until you make IT

- Simple idea, but very powerful
- VM`s on laptop model Cloud environment
- Reproducible Workflows
- Automation with Ansible, Vagrant & Packer
- And yes, you can use Docker too



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



Quickstart

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

cd vagransible

vagrant up centos6

Vagrant, Packer

- Builders
- Providers
- Provisioners
- Configuration

What do you need?

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

Provider agnostic

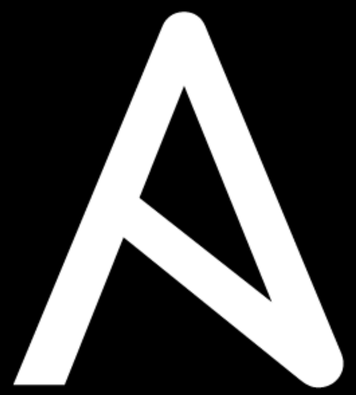
- VirtualBox
- VMWare
- Amazon Web Service
- Docker
- Microsoft Hyper-V
- …

Provisioner agnostic

- Ansible
- Salt
- Puppet
- Chef
- bash
- …

vagrant <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



vagrant up

environment

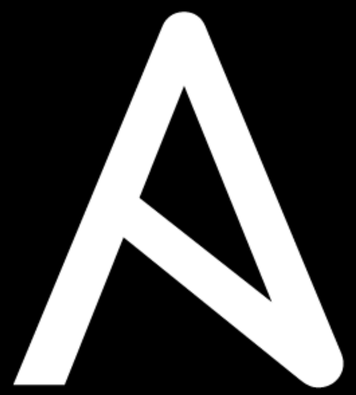
```
kreta:environment bas$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'chef/centos-6.5'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'chef/centos-6.5' is up to date...
==> default: Setting the name 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...
```

Vagrantfile

```
Vagrant.configure(2) do |config|

  config.vm.box = “dockpack/centos6"

end
```



provision with 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 selection

```
# in ~/.bash_profile

EXPORT VAGRANT_DEFAULT_PROVIDER=virtualbox
```

```
# in Vagrantfile

# Prefer VirtualBox before VMware Fusion

config.vm.provider "virtualbox"

config.vm.provider "vmware_fusion"
```

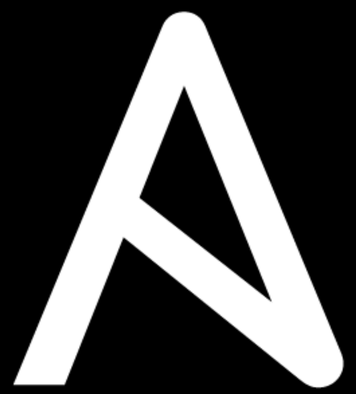
Building your own box

packer build packer-centos.json

kickstart install for RedHat-like systems.

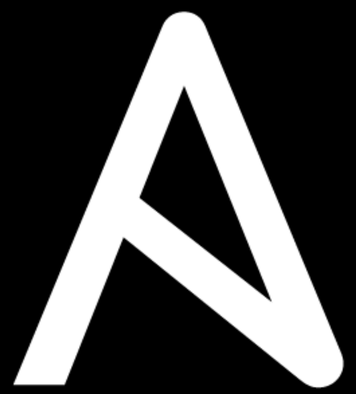
install ansible with a small shell script .

ansible does the rest in local mode.



provisioning in packer json

```
"provisioners": [
  {
    "type": "shell",
    "execute_command": "echo 'vagrant' | {{.Vars}} sudo -S -E bash '{{.Path}}'",
    "override": {
      "virtualbox-iso": {
        "scripts": [
          "scripts/ansible.sh"
        ]
      }
    }
  },
  {
    "type": "ansible-local",
    "playbook_file": "packer.yml",
    "role_paths": [
      "roles/bbaassssiiee.commoncentos",
      "roles/RHEL6-STIG"
    ]
  }
```



Resources

@bbaassssiiee

https://github.com/bbaassssiiee

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

https://galaxy.ansible.com

http://www.vagrantup.com