Ansible Ubuntu VM

Install, Vagrant, Ansible and Virtual Box.

$ mkdir ansible
$ cd ansible
$ vagrant init ubuntu/trusty64

$ vagrant up

Verify we are up with:

$ vagrant ssh

$ exit

$ vagrant ssh-config

Verify output:

HostName 127.0.0.1
    User vagrant
    Port 2222
    IdentityFile /scripts/ansible/.vagrant/machines/default/virtualbox/private_key

$ ssh vagrant@127.0.0.1 -p 2222 -i
/scripts/ansible/.vagrant/machines/default/virtualbox/private_key

^ Created an alias against above called **sshans** for quick ssh access.

$ exit

Create **hosts** file:

testserver ansible_ssh_host=127.0.0.1 ansible_ssh_port=2222 ansible_ssh_user=vagrant
ansible_ssh_private_key_file=/scripts/ansible/.vagrant/machines/default/virtualbox/private_key

Test connectivity with:

$ ansible testserver -i hosts -m ping

Add in ansable.cfg:

```
[defaults]
hostfile = hosts
remote_user = vagrant
private_key_file = /scripts/ansible/.vagrant/machines/default/virtualbox/priv
ate_key
host_key_checking = False
```

and update hosts to:

```
testserver ansible_ssh_host=127.0.0.1 ansible_ssh_port=2222
```

```
$ ansible testserver -m ping
```

```
$ ansible testserver -m command -a uptime
```

```
$ ansible testserver -a uptime
```

```
$ ansible testserver -a "tail /var/log/dmesg”
```

```
$ ansible testserver -s -a "tail /var/log/dmesg”

$ ansible testserver -s -m apt -a "name=nginx update_cache=yes”

$ ansible testserver -s -m service -a "name=nginx \ state=restarted"
```

———

```
$ vagrant halt

$ vagrant reload
```

Vagrantfile:

```
    VAGRANTFILE_API_VERSION = "2"
        Vagrant.configure(VAGRANTFILE_API_VERSION) do |config| config.vm.box
= "ubuntu/trusty64"
        config.vm.network "forwarded_port", guest: 80, host: 8080
        config.vm.network "forwarded_port", guest: 443, host: 8443
end
```

```
$ ansible-playbook web-notls.yml
```

Ansible sheebang line: #!/usr/bin/env ansible-playbook

make playbook executable: cmmod +x web-notls.yml

Then run like: ./web-notls.yml

YML Files:

Dictionaries, Lists, Line Folding, Plays, Modules

A Playbook is a list of dictionaries.

Every play must contain: A set of hosts to configure & A list of tasks to be executed.

Think of a play that connects hosts to tasks.

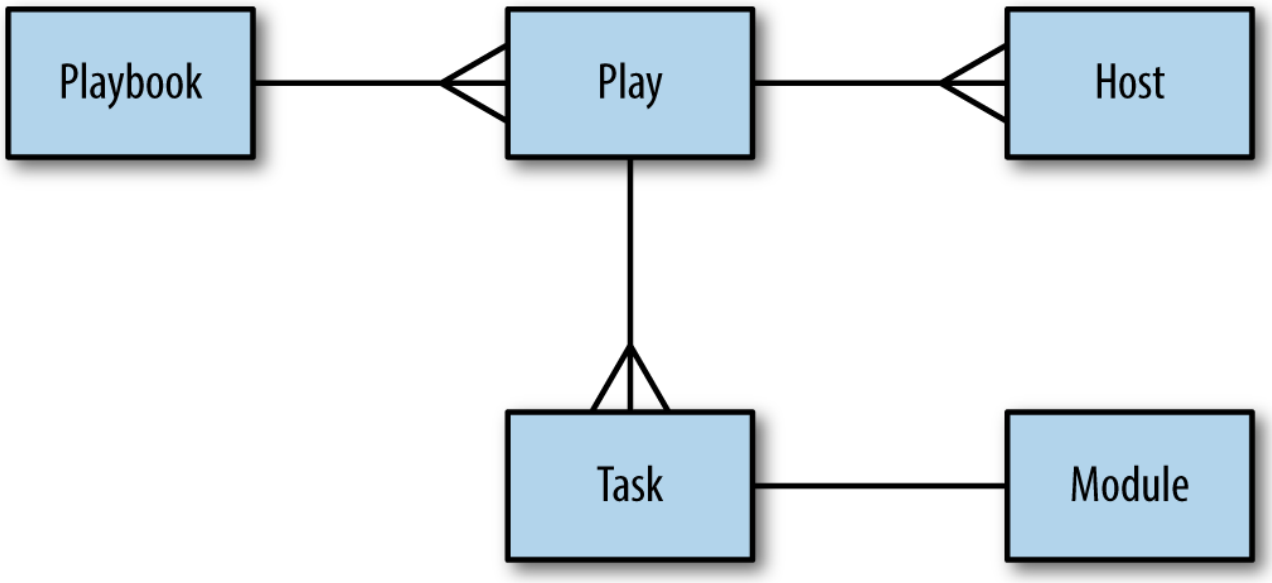
Dictionary: sudo: True - connects as root.

vars: A list of variables and values.

Out web-notes.yml has one play that has five tasks.

Modules used so far…: apt, copy, file, service.

ansible-doc - for ansible documentation like manages e.g: **ansible-doc service**



To manually generate a TLS certificate use:
openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -subj /CN=localhost -key out files/nginx.key -out files/nginx.crt

Vars:
e.g:

```
vars:
      key_file: /etc/nginx/ssl/nginx.key
      cert_file: /etc/nginx/ssl/nginx.crt
      conf_file: /etc/nginx/sites-available/default
      server_name: localhost
```

You reference variables using the **{{ braces }}** notation.

Handlers, trigger with notify task:

```
        notify: restart nginx
  handlers:
    - name: restart nginx
      service: name=nginx state=restarted
```