* A node can be both server and client
* No client installation needed
* PUSH method is used to send configuration written in playbooks
* Requirements : python and SSH
* There is order execution of task
* Based on yaml standard
* Daemonless

/usr/lib/python2.7/site-packages/ansible/modules

Syntax

--- # Our Favourite Movies

{Name: Yigit Aydog, Age:44, HireDate:01/05/1990}

Can detect datatypes (integer or float)

Ansible-doc -l : shows ansible modules and brief description

ansible-doc “module-name”

(location of ansible modules)

/usr/lib/python2.7/site-packages/ansible/modules

Server1: Control server-centos1

Server2: Web Client-centos2

Server3: Application Client-centos3

server1, server2, server3 (birbirleriyle ve kendi üzerinde şifresiz ssh’sız)

useradd test

passwd test

gpasswd -a test wheel

sudo visudo

%wheel ALL=(ALL) NOPASSWD: ALL

mkdir -p /home/test/playbooks/roles (or put it in network shared path so that all servers can access)

chown test:test /home/test/playbooks

yum install epel-release

yum update repolist

yum update

yum install ansible

ansible --version

yum install python-pip

pip install --upgrade pip

pip install passlib (for crypted password generation)

/etc/ansible/hosts: yönetilecek serverler (static inventory)

ansible all --list-hosts

ansible all -m ping

ansible “hostgroup” --list-hosts (hostgruptaki sunucuları listeler)

[local]

localhost ansible\_connection=local

[apacheweb] #ansible apacheweb -m ping (without override)

centos2

[appserver]

centos3

[testsrv]

testsrv ansible\_connection=ssh ansible\_user=test

[testsrv:vars]

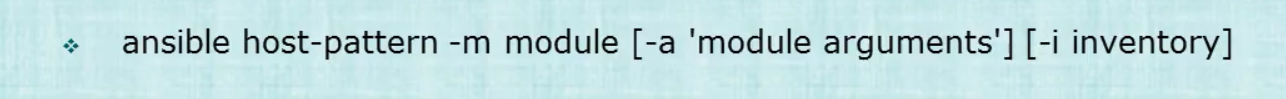
pac=packagename

ansible testsrv -m yum -a "name={{ pac }} state=latest" -b (passing variable from host file to playbook)

/etc/ansible/ansible.cfg : configurasyon dosyası

Ansible-config view

**Karşı sunucuda komut çalıştırmak (ad-hoc commands)**



ansible apacheweb -m shell -a 'cat /etc/fstab' -b -f <num of forks>

ansible apacheweb -m command -a 'cat /etc/fstab' -b

ansible apacheweb -m yum -a "name=elinks state=latest" -b

ansible apacheweb -b -a "touch /home/yaydog/testfile" --become-user yaydog

(karsı tarafın /home/yaydog altına dosya oluşturur ve owner’ı yaydog olur)

(about forking)

* Each fork is able to run a task against a target host.
* By default, Ansible executes 5 forks.
* If more than 5 hosts are targeted, hosts after the first 5 must wait for a fork to become free before being able to execute.
* The number of forks used by Ansible may be configured in ansible.cfg or on an as-needed basis using the -f or --forks flag.

(modules)

ansible-doc -l : list all available modules and with brief description

**(SETUP MODULE)**

**Getting node settings-system facts**

ansible hostname -m setup --tree /tmp/facts (creates system facts file)

ansible hostname -m setup (display on console)

ansible hostname -m setup -a 'filter=ansible\_interfaces'

common system facts:

* ansible\_fqdn
* ansible\_interface
* ansible\_kernel
* ansible\_memtotal\_mb
* ansible\_proc\*
* ansible\_virt\*
* ansible\_architecture
* ansible\_distribution
* ansible\_domain

(filtering for detailed info)

- debug: msg="interface alias is {{ ansible\_default\_ipv4.alias }}"

**FILE MODULE**

ansible apacheweb -m file -a 'path=/etc/fstab' (dosya yada klasör özelliklerini getirtir)

ansible apacheweb -m file -a 'path=/home/test/asd state=directory owner=root' -b (dosya yada klasör özelliklerini değiştirmek)

ansible apacheweb -m copy -a 'src=/etc/fstab dest=/tmp/' -b

**İnstalling packages on remote server**

ansible apacheweb -m yum -a ‘pkg=lynx state=installed update\_cache=true’

**Creating playbook and install multiple packages**

cd /home/test/playbooks

touch appserver.yml

--- #test playbook

- hosts: apacheweb

vars:

http\_port: 80

control\_server: localhost

webroot: /var/www/html

tasks:

- name: install programs

yum: name=httpd state=installed update\_cache=true

ansible-playbook --syntax-check appserver.yml (syntax check)

ansible-playbook appserver.yml --check (dry run)

ansible-playbook appserver.yml -b (real run)

ansible-playbook appserver.yml -b -vvv (increased verbosity)

ansible-playbook appserver.yml -b -l centos2 (inventory de olsa dahi diğer sunucularda çalıştırmaz)

(run playbook for dynamic inventory-script must be executable)

ansible-playbook appserver.yml -b -i /etc/ansible/inventory.py

(take inventory information and run playbook accordingly)

ansible-playbook appserver.yml -b -i /etc/ansible/hosts

**encrypted password generation for user module**

python -c "from passlib.hash import sha512\_crypt; import getpass; print sha512\_crypt.using(rounds=5000).hash(getpass.getpass())"

**SECURING YAML FILE –creating vault**

Creating>> ansible-vault create secure.yml

Editing>> ansible-vault edit secure.yml

Changing vault password>> ansible-vault rekey secure.yml

Decrypt>> ansible-vault decrypt secure.yml

Encrypt>> ansible-vault encrypt secure.yml

Running vault file>> ansible-playbook appserver.yml -b --ask-vault-pass

**Error handling**

ignore\_errors: yes/no

**Starting from specific task**

ansible-playbook appserver.yml -b --start-at-task='install client software'

(belirli bir tasktan sonra başlar ve o taskı yapar ve ondan sonraki taskları da yapar)

ansible-playbook appserver.yml -b --step

(taskları Y/N şeklinde sorarak yapar)

**USING HANDLERS**



**USING TAGS(PLAYBOOKTAKI SADECE TAG İLE BELİRLENMİŞ GOREVİ YAPAR)**

- name: Add a multiple line to a file

blockinfile:

path: /etc/hosts

block: |

192.168.1.99 testsrv

192.168.1.100 testsrv2

tags:

- hostfile

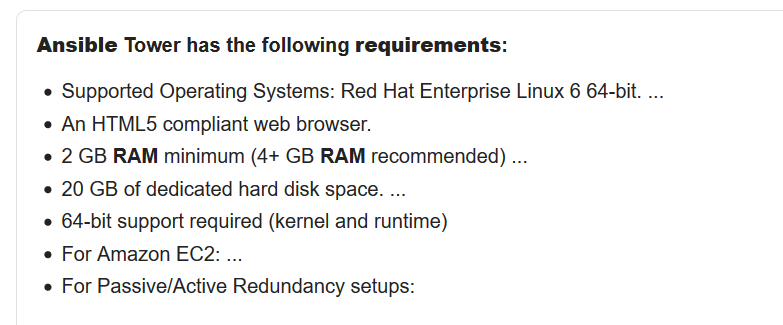
ansible-playbook appserver.yml -b --tags "hostfile"

**Getting variable from console and use it in playbook**

playbook

user: '{{ user }}'

ansible-playbook appserver.yml -b --extra-vars "user=test"



**ansible tower installation**

<https://computingforgeeks.com/install-and-configure-ansible-tower-on-centos-rhel/>

username: admin

password: defined in configuration file

<https://192.168.1.50/#/login>

**ansible tower notes**

Stopping ansible tower: ansible-tower-service stop

**Ansible and windows**

Ansible requires PowerShell 3.0 or newer and at least .NET 4.0 to be installed on the Windows host.

A WinRM listener should be created and activated

OpenSSH must be installed

Powershell version: (Get-Host).Version

.NET Framework version > HKLM: Sofware: Microsoft: Net Framework Setup: NDP: v4: Full

Check winrm is working: winrm enumerate winrm/config/listener

<https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html#host-requirements>

<https://github.com/PowerShell/Win32-OpenSSH/wiki/Install-Win32-OpenSSH>

*\*on windows machine*

Upgrading PowerShell and .NET Framework.ps1

ConfigureRemotingForAnsible.ps1

*\*on ansible controller*

**packages**

yum install python-pip

pip install --upgrade pip

pip install pywinrm

**host file**

[windows]

192.168.1.10

[windows:vars]

ansible\_user=administrator

ansible\_password=Lapland23\*

ansible\_connection=winrm

ansible\_winrm\_server\_cert\_validation=ignore

(testing)

ansible windows -i hosts -m win\_ping

ansible windows -m setup

(running playbook on windows)

ansible-playbook appserver.yml