

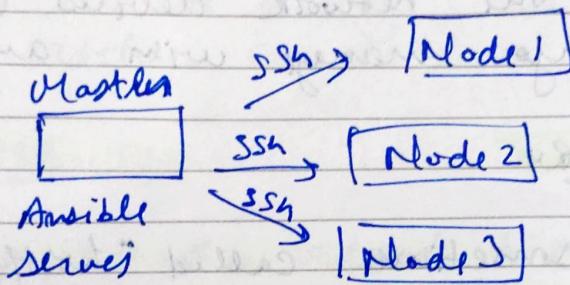
ANSIBLE

Ansible

Ansible is an open-source IT engine/ configuration management tool that automates application deployment, cloud provisioning, intra service orchestration and other IT tools.

- It uses YAML scripting
- works on push management
- Light weight and consistent.
- easy to install
- Free and opensource
- It ~~uses~~ is agentless and communicates through SSH.
- OpenSSH security features makes it very secure.

→ One master can control configuration of many servers



IMPORTANT TERMINOLOGY

(i) Control Node / Master Node

Commands and playbooks can run by invoking /usr/bin/ansible or /usr/bin/ansible-playbook from any control node.

You can use any computer that has Python installed on it as a control node.

(ii) Ansible Server

It is the machine where Ansible is installed.

Playbooks can run here.

(iii) MANAGED NODES

Also sometimes called 'hosts', managed nodes are the network devices (and/or servers) you manage with ansible.

(iv) INVENTORY

Also sometimes called "hostfile", inventory is the list of managed nodes used to organize them. It is also used for creating and nesting groups for easier scaling.

It is a file containing data about client servers.

(v) MODULES

These are the units of code executed by ansible. Each module can be used for a specific purpose.

We can invoke a single module with a task, or invoke several different modules in a playbook.

Command or set of commands to be executed by client side.

(vi)

TASKS -

The units of action in Ansible. One can execute a single task once with an ad-hoc command.

(vii)

PLAYBOOKS

These are the ordered list of tasks that are saved so you can run those tasks in that order repeatedly.

Playbooks are written in YAML and are easy to read, write, share and understand.

(viii)

PLAY

The execution of a Playbook.



Inventory file location

→ /etc/ansible/hosts



Ad-hoc commands can be performed individually to perform a quick function.

CREATING Ansible Inventory/hosts FILE

/etc/ansible/hosts → default inventory file location

if no such file or directory found
then create the directory and file.

→ mkdir ansible

then now

→ cd ansible

then create the hosts file

→ vim hosts

Inside Inventory file

[servers]

server1 ansible_host = IP Address

server2 ansible_host = IP Address

servern n n

[all:vars]

ansible -bpython -i inventory >/usr/bin/python3
from save it

↳ this will make sure every server has same python

host file is created.

to check the inventory file is correct or not.

→ ansible-inventory --list -y

↳ if the host file is in default location

→ ansible-inventory --list -y -i /home/ubuntu/ansible/hosts
↳ path to host file.

Ansible Commands

i) ansible all -m ping -i inventory --private-key = /v.ssh/key
all servers
↳ path to key

→ changed 600 key file
↳ so that command can access the file.

(ii) ansible all -a "free -h" -i inventory path

-private-key = key path

to check the RAM
disk space of all the
servers

for disk space use "free df -h"

(iii) ansible servers -a "uptime" -i inventory path

-private-key = key path

to check the uptime of servers

ANSIBLE PlayBook Example

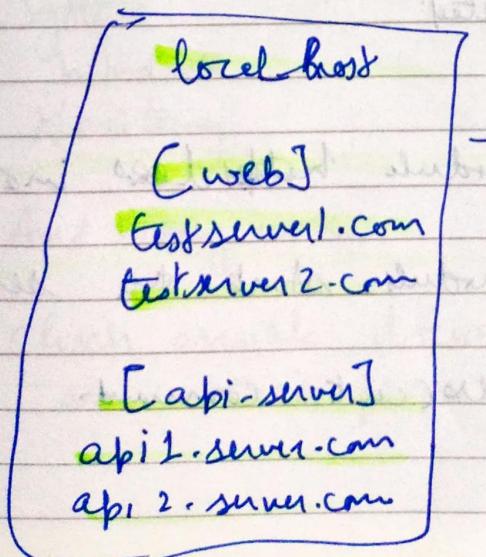
- It written in YAML
- Playbook defines set of plays with activities to run on host
- A task is a single action perform on a host

-
name : Play1
hosts : localhost
tasks :

- name : "checking date"
command : date
- name : "installing httpd"
yum:
 - name: httpd
 - state: present
- name: "start httpd server"
service!
 - name: httpd
 - state: started

→ Mapping host with Playbook.

In inventory file.



→ inventory file

we can replace localhost with web or api-server to map and execute that.

PLAYBOOK Module Examples

-

```
name: Play 1
hosts: localhost
tasks:
  - name: "checking date"
    command: date
```

-

```
name: Play 2
hosts: localhost
tasks:
  - name: "installing httpd"
    yum:
      name: httpd
      state: present
  - name: "start httpd server"
    service:
      name: httpd
      state: started
```

Yum → this module ~~not~~ does installation

Service → this module starts the service

Command → to execute command

PlayBook Execution

Ansible-playbook filename.yml -i inventory file name
to execute the playbook

ANSIBLE MODULES

Ansible modules are reusable, standalone scripts that can be used by the Ansible API, or by other Ansible or ansible-playbook programs.

Some of the modules:

- (i) Cloud → cloud module works with all cloud providers
- (ii) Clustering
- (iii) Command
- (iv) File
- (v) Database
- (vi) System
- (vii) Windows
- (viii) Monitoring

And many more.

Check ansible documentation for modules and their properties.

ANSIBLE CONDITION

Example:

```
- name: httpd Install
```

```
hosts: all
```

```
tasks:
```

```
- name: install httpd on centos
```

```
yum!
```

```
name: httpd
```

```
state: present
```

```
condition ← [ when: ansible-os-family == "centos" ] ✓
```

```
ansible-os-family == "SCL"
```

```
- name: install httpd on redhat
```

```
apt:
```

```
name: httpd
```

```
state: present
```

```
condition ← [ when: ansible-os-family == "debian" ] ✓
```

```
ansible-os-family == "Ubuntu"
```

Ansible Loop

Example

```
- name: httpd install  
  hosts: all
```

vars:

packages:

```
- name: httpd
```

required: false

```
- name: mysql
```

required: true

```
- name: apache
```

required: true

tasks:

```
- name: install httpd on centos
```

yum:

```
  name: "{{ item.name }}"
```

state: present

```
when: item.required == True
```

```
loop: "{{ packages }}"
```