BATCH: 149

04/10/2023

DT/NT : DT

LESSON : DevOps

SUBJECT: Ansible 2

Playbooks


techproeducation.com


+1 (585) 304 29 59

TECHPRO EDUCATION

- `ANSIBLE_CONFIG` (environment variable if set)
- `ansible.cfg` (in the current directory)
- `~/.ansible.cfg` (in the home directory)
- `/etc/ansible/ansible.cfg`

ansible.cfg

ansible4 > ansible.cfg

```
[defaults]
host_key_checking = False
inventory = inventory.txt
deprecation_warnings=False
interpreter_python=auto_silent

```

inventory

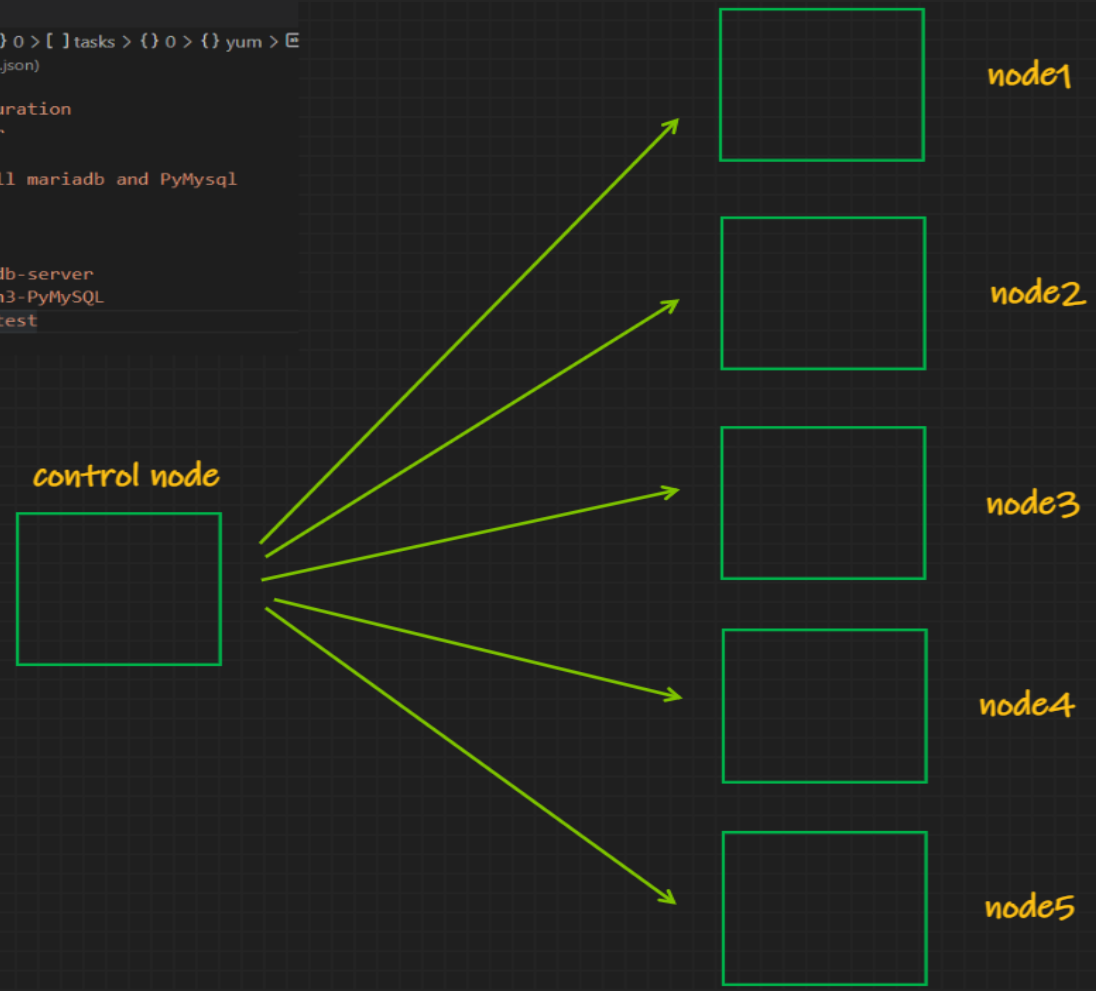
```
$ vim hosts
[webservers]
node1 ansible_host=<node1_ip> ansible_user=ec2-user
node2 ansible_host=<node2_ip> ansible_user=ec2-user

[all:vars]
ansible_ssh_private_key_file=/home/ec2-user/<pem file>

[devservers]
node1

[testgroup:children]
webservers
devservers

[dev_sub_gr]
#node[1:20]
node[1:2]
```

playbook.yml 4

ansible4 > playbook.yml > {} 0 > [ ] tasks > {} 0 > {} yum >

Ansible Playbook (ansible.json)

```
---
- name: db configuration
  hosts: db_server
  tasks:
    - name: install mariadb and PyMysql
      become: yes
      yum:
        name:
          - mariadb-server
          - python3-PyMySQL
        state: latest

```



Manager

Address List

Hillroad Road, 43

Oxford Square, 15

Checklist

ANSI

Faringdon Road, 4

Roman Walk, 9

Control Node

Inventory

Hillroad Road, 43

Oxford Square, 15

Playbook

Ansible

Faringdon Road, 4

Roman Walk, 9

Playbooks

inventory.ini

```
[webservers]
node1 ansible_host=54.174.120.241
node2 ansible_host=3.84.254.65

[databases]
node3 ansible_host=54.174.102.205
```

playbook.yml

play-1

```
---
- name: update web servers
  hosts: webservers
  remote_user: root

  tasks:
  - name: ensure apache is at the latest version
    yum:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf
```

play-2

```
- name: update db servers
  hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is at the latest vers
    yum:
      name: postgresql
      state: latest
  - name: ensure that postgresql is started
    service:
      name: postgresql
```

Playbooks

inventory.ini

```
[webservers]
node1 ansible_host=54.174.120.241
node2 ansible_host=3.84.254.65

[databases]
node3 ansible_host=54.174.102.205
```

How to Run

```
$ ansible-playbook playbook.yml
```

- **Playbooks** contain **plays**
- Plays contain **tasks**
- Tasks call **modules**

- **Tasks** run **sequentially**

playbook.yml

```
---
- name: update web servers
  hosts: webservers
  remote_user: root

  tasks:
  - name: ensure apache is at the latest version
    yum:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf

- name: update db servers
  hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that postgresql is started
    service:
      name: postgresql
```

play-1

play-2

modules

arguments

arguments

arguments

arguments

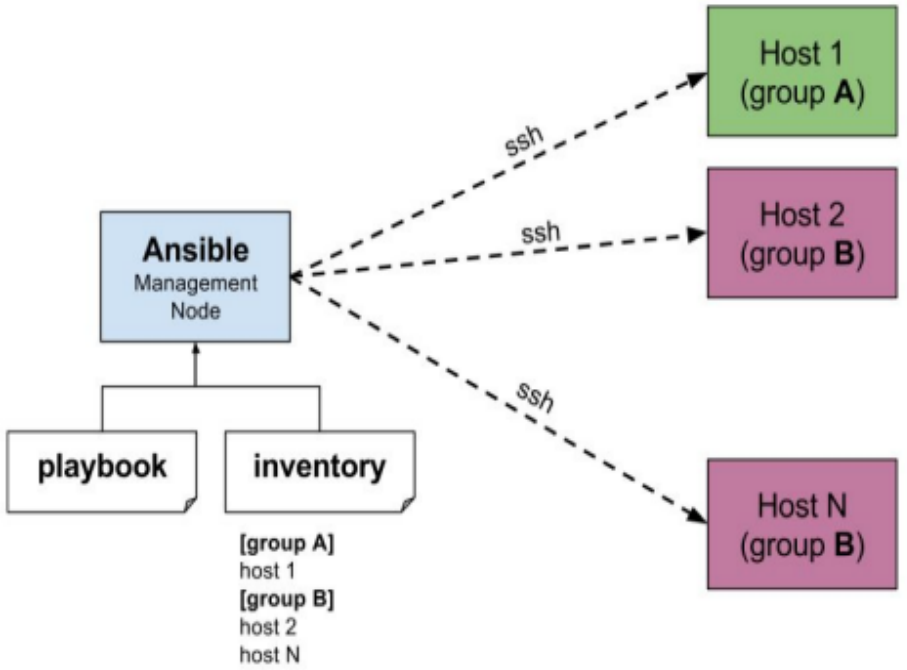
task-1

task-2

task-1

task-2

Hosts and Users

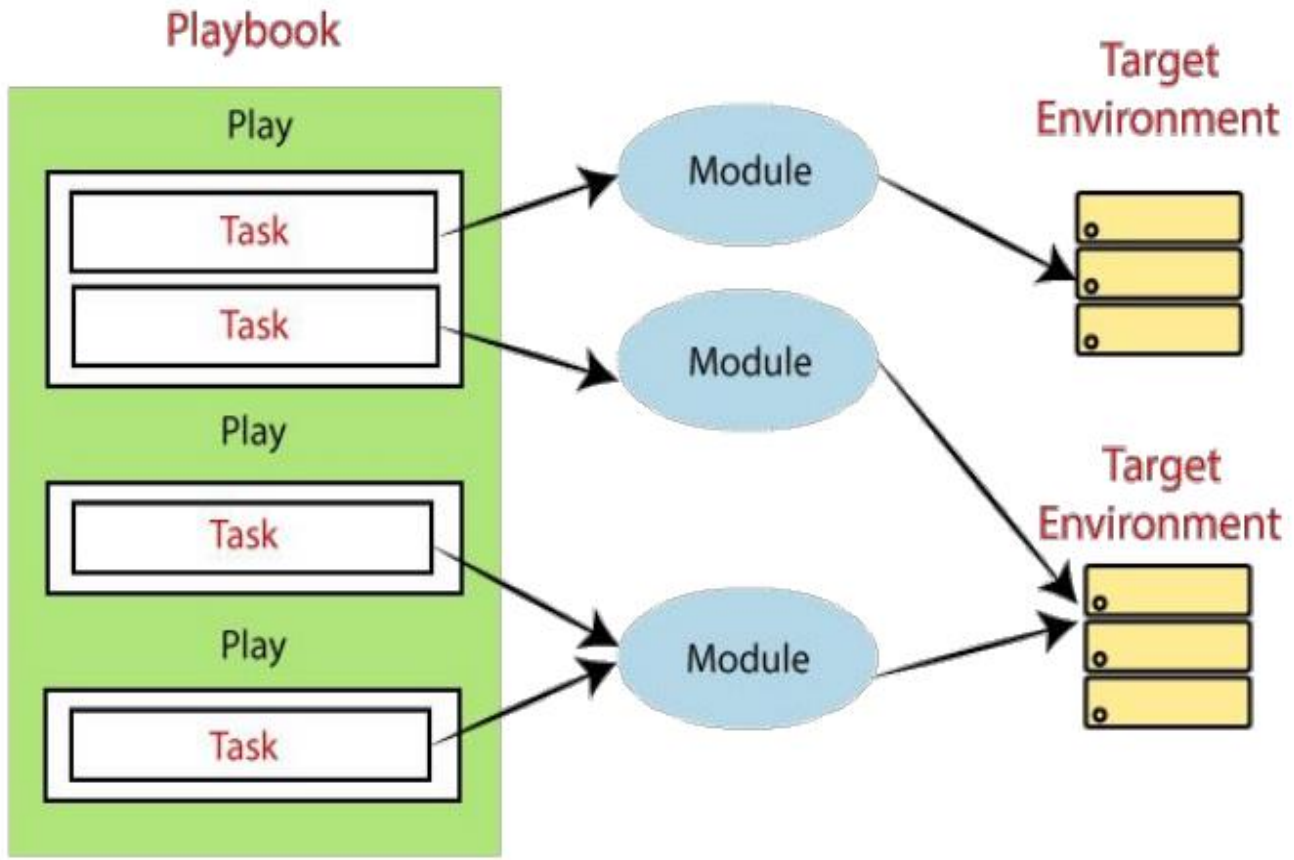


Tasks

- Each play contains a list of tasks. Tasks are executed in order, one at a time, against all machines matched by the host pattern, before moving on to the next task.
- The goal of each task is to execute a module, with very specific arguments. Variables can be used in arguments to modules.

```
# Simple Ansible Playbook1.yml
-
  name: Play 1
  hosts: localhost
  tasks:
    - name: Execute comand "date"
      command: date

    - name: Execute script on server
      script: test.sh

    - name: Install httpd package
      yum:
        name: httpd
        state: present

    - name: Start web server
      service:
        name: httpd
        state: started
```

Modules


Playbook
Play
Task
Task
Play
Task
Play
Task
Module
Module
Module
Target Environment
Target Environment

Modules

- **Modules** (also referred to as “task plugins” or “library plugins”) are discrete units of code that can be used from the command line or in a playbook task.
- Ansible executes each module, usually on the remote target node, and collects return values.
- Modules should be **idempotent**, and should avoid making any changes if they detect that the current state matches the desired final state.

playbook.yml

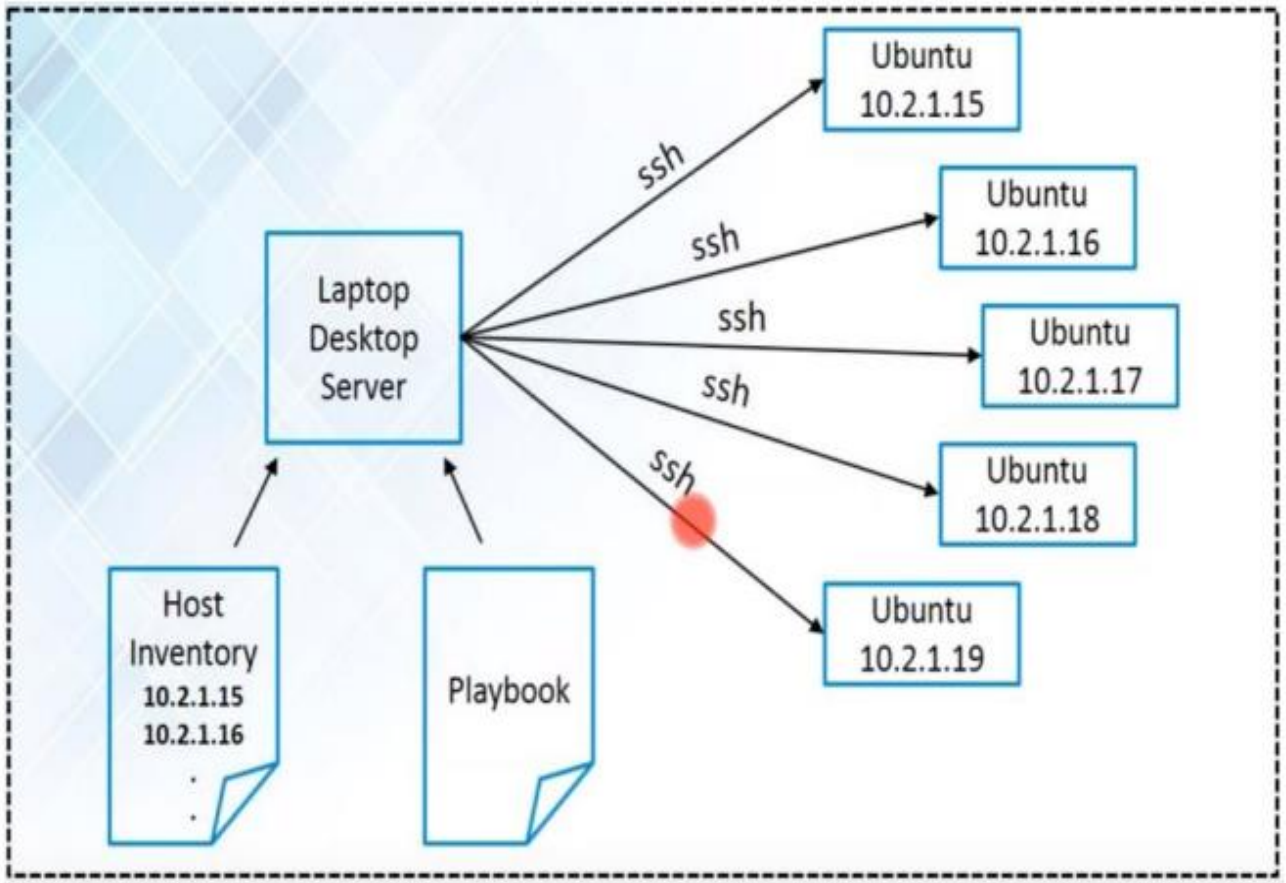
```
-
  name: Play 1
  hosts: localhost
  tasks:
   - name: Execute command 'date'
     command: date

   - name: Execute script on server
     script: test_script.sh

   - name: Install httpd service
     yum:
      name: httpd
      state: present

   - name: Start web server
     service:
      name: httpd
      state: started
```

Handlers

Handlers are lists of tasks, not really any different from regular tasks, that are referenced by a globally unique name, and are notified by notifiers. If nothing notifies a handler, it will not run.

```
- hosts: webservers1
  user: root
  tasks:
  - name: test copy
    copy: src=/root/a.txt dest=/mnt
    notify: test handlers
  handlers:
  - name: test handlers
    shell: echo "abcd" >> /mnt/a.txt
~
```

Inventory File



Inventory File

- Ansible works against multiple managed nodes or “hosts” in your infrastructure at the same time, using a list or group of lists know as inventory.
- The default location for inventory is a file called /etc/ansible/hosts.
- You can specify a different inventory file at the command line using the -i < path > option.

Inventory Files

```
$ app.inv
[webservers]
www1.example.com
www2.example.com

[appservers]
app1.example.com
app2.example.com

[memcached]
memcached.example.com

[redis]
redis.example.com

[dbservers]
db0.example.com
```