Ansible Playbook Basics

1. To create an EC2 instance – ubuntu, t2 micro with the below user data,

#!/bin/bash

sudo apt update

sudo apt install software-properties-common -y

sudo add-apt-repository --yes --update ppa:ansible/ansible

sudo apt install ansible –y

Make sure to create a new security group which allows port 22 from own ip, and new key pair.

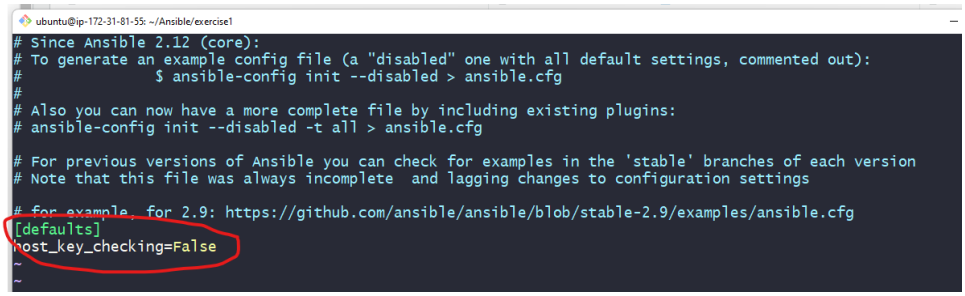
2. To create 2 EC2 instances for web tier, - centos, t2 micro. Create a new SG and make sure to allow port 22 connection from Ansible security group and own ip. Also create a new key file.
3. To create a EC2 instance for Db tier, create a new sg and make sure to allow port 22 connection from Ansible security group and own ip. Can use the above sec key file.
4. Login to Ansible VM and create /Ansible/exercise1 directories

```
buntu@ip-172-31-81-55:~$ pwd
home/ubuntu
buntu@ip-172-31-81-55:~$ mkdir Ansible
buntu@ip-172-31-81-55:~$ cd Ansible/
buntu@ip-172-31-81-55:~/Ansible$ mkdir exercise1
buntu@ip-172-31-81-55:~/Ansible$ cd exercise1/
buntu@ip-172-31-81-55:~/Ansible/exercise1$ ansible --version

nsible [core 2.13.5rc1]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/us
/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ans
ble/collections
  executable location = /usr/bin/ansible
  python version = 3.10.4 (main, Jun 29 2022, 12:14:53) [GCC 11.2.0]
  jinja version = 3.0.3
  libyaml = True
buntu@ip-172-31-81-55:~/Ansible/exercise1$
buntu@ip-172-31-81-55:~/Ansible/exercise1$
```

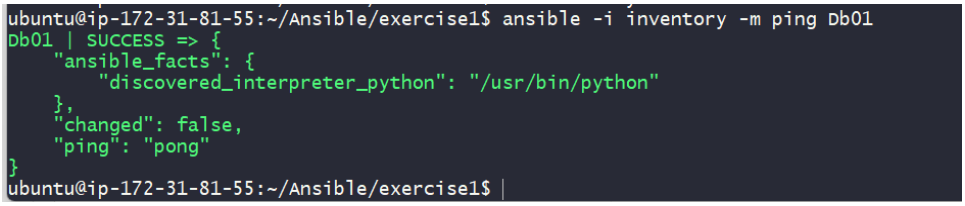
5. Create an inventory file with all the host information.

```
Web01 ansible_host=172.31.95.159 ansible_user=centos ansible_ssh_private_key_file=remote.pem
Web02 ansible_host=172.31.94.72 ansible_user=centos ansible_ssh_private_key_file=remote.pem
Db01 ansible_host=172.31.88.160 ansible_user=centos ansible_ssh_private_key_file=remote.pem

[Webvm]
Web01
Web02

[Dbvm]
Db01
```

6. To create a remote.pem file in the same directory and paste the private key. And chmod 400 remote.pem for file permission.
7. To add the below highlighted line in “sudo vi /etc/ansible/ansible.cfg”



8. Execute below command to check the remote host connection, similarly for Web01 and Web02.



9. We can check for webvm and DB vm groups

```
ubuntu@ip-172-31-81-55:~/Ansible/exercise1$ ansible -i inventory -m ping Webvm
web01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
web02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-81-55:~/Ansible/exercise1$ ansible -i inventory -m ping Dbvm
Db01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-81-55:~/Ansible/exercise1$
```

10. Inventory file can be written with variable definition,

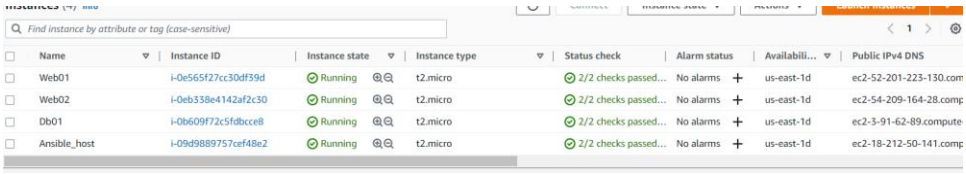
```
Web01 ansible_host=172.31.95.159
Web02 ansible_host=172.31.94.72
Db01 ansible_host=172.31.88.160

[Webvm]
Web01
Web02

[Dbvm]
Db01

[dc_ohio:children]
Webvm
Dbvm

[dc_ohio:vars]
ansible_user=centos
ansible_ssh_private_key_file=remote.pem
```

11. To ping all the host in the inventory file,

    ansible -i inventory –m ping '*'

Exercise 2:

Created 4 VM in EC2, one to host ansible, two for web apps and one for DB



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabili...	Public IPv4 DNS
Web01	i-0e565f27cc30df39d	Running	t2.micro	2/2 checks passed...	No alarms	us-east-1d	ec2-52-201-223-130.com
Web02	i-0eb338e4142af2c30	Running	t2.micro	2/2 checks passed...	No alarms	us-east-1d	ec2-54-209-164-28.comp
Db01	i-0b609f72c5fdbcce8	Running	t2.micro	2/2 checks passed...	No alarms	us-east-1d	ec2-3-91-62-89.compute-
Ansible_host	i-09d9889757cef48e2	Running	t2.micro	2/2 checks passed...	No alarms	us-east-1d	ec2-18-212-50-141.comp

For ansible machine, add the below user data,

```
#!/bin/bash
sudo apt update
sudo apt install software-properties-common –y
sudo apt-add-repository –yes –update ppa:ansible/ansible
sudo apt install ansible -y
```

Have the below files in your directory,

ubuntu@ip-172-31-81-55: ~/Ansible/exercise3

```
buntu@ip-172-31-81-55:~/Ansible/exercise3$ ls
ndex.html  inventory  remote.pem  web_db.yaml
buntu@ip-172-31-81-55:~/Ansible/exercise3$
```

Index.html - a basic html file.

Inventory file with below data,

```
Web01 ansible_host=172.31.95.159
Web02 ansible_host=172.31.94.72
Db01 ansible_host=172.31.88.160

[Webvm]
Web01
Web02

[Dbvm]
Db01

[dc_ohio:children]
Webvm
Dbvm

[dc_ohio:vars]
ansible_user=centos
ansible_ssh_private_key_file=remote.pem
~
~
~
```

The below playbook will install web servers on WebVM and db server on db vm and then move the index.html file from local to remote web vm in /var/www/html directory and make a backup of older version of the file.

web_db.yaml file will be looks like below,

```
---
- name: Setup WebServer
  hosts: Webvm
  become: yes
  tasks:
    - name: Install Apache httpd
      yum:
        name: httpd
        state: present
    - name: Start & Enable HTTPD
      service:
        name: httpd
        state: started
        enabled: yes
    - name: Deploy web file
      copy:
        src: index.html
        dest: /var/www/html/index.html
        owner: root
        group: root
        mode: '0644'
        backup: yes

- name: Setup DBserver
  hosts: Dbvm
  become: yes
  tasks:
    - name: Install MySQL server
      yum:
        name: mariadb-server
        state: present
    - name: Start & Enable mariadb service
      service:
        name: mariadb
        state: started
        enabled: yes
```

Next step is to create a database in Maria SQL db and the user. So Play book will be updated.

- name: Create a new database with name 'accounts'

mysql_db:
name: accounts
state: present

Before adding this, there is a dependency on installing python my sql.

Login to db machine and search below

yum search python | grep -i mysql

So **MySQL-python** is the package, so add another task in playbook,

- name: Install Python MySQL
  yum:
  name: MySQL-python
  state: present

Add a new user task now in playbook,

- name: Create db user
  mysql_user:
  name: bob
  password: 12345
  priv: '*.*ALL'
  state: present

Exercise 3: Solution to create EC2 Key pair and EC2 instances in AWS

Prerequisites:
The playbook running on Ansible hosts should be aware of AWS creds.
Create a user and get the creds (access key) in AWS.
Refer: https://docs.ansible.com/ansible/latest/collections/amazon/aws/docsite/guide_aws.html

Add the below lines in bashrc file,

```
export AWS_ACCESS_KEY_ID='AK123'
export AWS_SECRET_ACCESS_KEY='abc123'
```

Then source .bashrc, the file will execute and update bashrc(I.e the env variable will be updated with new values for secret key)

We need to install pip, boto and boto3.

```
- hosts: localhost
  gather_facts: False
  tasks:
    - name: Create key pair
      ec2_key:
        name: sample
        region: us-east-1
      register: keyout


      #- name: print key
      #debug:
      #  var: keyout
```

```
- name: save key
  copy:
    content: "{{keyout.key.private_key}}"
    dest: ./sample.pem
  when: keyout.changed


- ec2:
    key_name: sample
    instance_type: t2.micro
    image: ami-0f9fc25dd2506cf6d
    region: us-east-1
    wait: yes
    exact_count: 1
    instance_tags:
       Name: db01
       db: postgres
    count_tag:
       Name: db01
       db: postgres
```