

Name: Chaudhari Ganesh Dadabhau

Integration of Ansible with Docker

Ansible TASK 1 :

Write an Ansible playbook to perform following operations in managed nodes

- Configure yum repository
- Configure Docker on top of managed node
- Start docker services
- configure Apache server(httpd) in docker with public access

Ansible :

Ansible is tool or framework which is used for automation. It works on push mechanism. we only need to tell the Ansible " what we want" not " how to do" so that it Declarative language. It is python library. Ansible only used for configuration purpose. it follow Idempotance rule.

Docker :

Docker is used to manage lifecycle of container. It is a tool designed to make it easier to create, deploy, and run applications by using containers. It provides isolated environment for docker images .

Setup of Ansible :

The system on which ansible setup is located is called controller node and the target system is called managed node. First we need to install ansible on controller node using pip3 because it is one of the python library and to check version of ansible we need to use `ansible --version`.

```
# pip3 install ansible
```

```
# ansible --version
```

We need to create inventory file which tell the ansible about target systems here **/etc/myhost.txt** is inventory file which contains the details of target machine like ip, username and password or public , private key with password. here i used ip and username ,password of target machine.

after that ansible must know the path of inventory file for that we need to create `/etc/ansible/ansible.cfg` and put path of inventory file.

```
192.168.225.27 ansible_ssh_user=root ansible_ssh_pass=redhat

[defaults]
inventory = /etc/myhost.txt
host_key_checking = False # it is false because host key checking doesnot support sshpass
```

For establishing connection between controller node and managed node we need to install **sshpass** on **controller node** but it is not present in yum repository then we need to download and install **epel-release-latest-8.noarch.rpm** . After installation ansible setup is completed and now you can check connection by using following command .

```
[root@localhost ~]# ansible all -m ping
192.168.225.27 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
```

Integration of Ansible with Docker

1.Configuration of yum repository and installation of docker: we need configure yum repository because it helps for installation of docker. Then configure yum repository and install docker on top of managed node using YAML modules.

```

File Edit View Search Terminal Help
name: Integration of ansible with docker
hosts: all

tasks:
  # configuration of repository
  - name: Adding repolist for docker installation
    yum_repository:
      name: "docker"
      description: "docker repository setup"
      baseurl: "https://download.docker.com/linux/centos/7/x86_64/stable"
      gpgcheck: no

  # Installation of docker
  - name: installation of docker
    package:
      name: "docker-ce-18.09.1-3.el7.x86_64"
      state: present

```

2. Start the docker service, create directory and copy content on managed node: To access docker we need to start docker service, create folder **/ganesh** because we want to transfer data of that folder from managed node to server. and copy content from controller node to managed in **/ganesh/index1.html** folder so that we can transfer data on server. To do that use following modules

```

# It is used to start docker service

- service:
  name: "docker"
  state: started

# Create directory

- name: create directory
  file:
    path: /ganesh
    state: directory

# Copying File to Managed node from Control node

- name: copying directory
  copy:
    content: "Ansible is only used for configuration purpose and it is python library \n"
    dest: /ganesh/index1.html

```

3. Start firewalld service : If firewalld is closed then container installation will not be taken so start firewalld service using service module.

```

# used to start firewalld service which is necessary for port mapping
- name: used to start firewalld
  service:
    name: firewalld
    state: started

```

4. Pull and launch the httpd server container: To run container docker image is required for that it pull image from docker hub if image is not locally available. After we need to open port 80 of container and map port 4545 of managed node with port 80 of container using following modules.

```
# It launch the container for that it find image on host machine if it is not available then it pull from docker hub.
- name: configuration of server
  docker_container:
    name: server
    image: httpd
    state: started
    exposed_ports:
      - 80
    ports:
      - 4545:80
    volumes:
      - /ganesh:/usr/local/apache2/htdocs
```

5. Stop firewalld and watch the container: Firewall should be closed because it won't give the access of container to the world through 4545 port of managed node. and watch container details using command module.

```
# It is used to stop firewalld because to access container outside world through port 4545.
- name: used to stop firewalld
  service:
    name: firewalld
    state: stopped

# It is used to watch details of container
- name: used to watch details of container
  command: "docker inspect server"
  register: x

# used to watch the details of container
- debug:
  var: x
```

To run container use :

ansible-playbook docker.yml

After running docker.yml see output:

```
[root@localhost playbook]# ansible-playbook docker.yml
PLAY [Integration of ansible with docker] *****
TASK [Gathering Facts] *****
ok: [192.168.225.27]
TASK [Adding repolist for docker installation] *****
ok: [192.168.225.27]
TASK [installation of docker] *****
ok: [192.168.225.27]
TASK [service] *****
ok: [192.168.225.27]
TASK [create directory] *****
ok: [192.168.225.27]
TASK [copying directory] *****
ok: [192.168.225.27]
```

```

TASK [used to start firewall] *****
changed: [192.168.225.27]

TASK [configuration of server] *****
[WARNING]: The value 80 (type int) in a string field was converted to '80' (type string). If this does not look like what you expect, quote
the entire value to ensure it does not change.
ok: [192.168.225.27]

TASK [used to stop firewall] *****
changed: [192.168.225.27]

TASK [used to watch details of container] *****

TASK [debug] *****
ok: [192.168.225.27] => {
  "x": {
    "changed": true,
    "cmd": [
      "docker",
      "inspect",
      "server"
    ],
    "delta": "0:00:00.156382",
    "end": "2020-08-05 11:24:11.353324",
    "failed": false,
    "rc": 0,
    "start": "2020-08-05 11:24:11.196942",
    "stderr": "",
    "stderr_lines": [],
    "stdout": "\nEndpointID\\": \\\"38b95b4b165b75a63edd32690e9c495a475ebe32c50d64eab78aaf097835d399\\\",",
    "stdout_lines": [
      "\nEndpointID\\": \\\"38b95b4b165b75a63edd32690e9c495a475ebe32c50d64eab78aaf097835d399\\\",",
      "\nGateway\\": \\\"172.17.0.1\\\",",
      "\nGlobalIPv6Address\\": \\\"\\\",",
      "\nGlobalIPv6PrefixLen\\": 0,",
      "\nIPAddress\\": \\\"172.17.0.2\\\",",
      "\nIPPrefixLen\\": 16,",
      "\nIPv6Gateway\\": \\\"\\\",",
      "\nMacAddress\\": \\\"02:42:ac:11:00:02\\\",",
      "\nNetworks\\": {",
      "  \"bridge\\": {",
      "    \"IPAMConfig\\": null,",
      "    \"Links\\": null,",
      "    \"Aliases\\": null,",
      "    \"NetworkID\\": \\\"cd762659afab6e02a9b8bbc9d163725e8f422f0bbcf582253644a9e2f15aa31\\\",",
      "    \"EndpointID\\": \\\"38b95b4b165b75a63edd32690e9c495a475ebe32c50d64eab78aaf097835d399\\\",",
      "    \"Gateway\\": \\\"172.17.0.1\\\",",
      "    \"IPAddress\\": \\\"172.17.0.2\\\",",
      "    \"IPPrefixLen\\": 16,",
      "    \"IPv6Gateway\\": \\\"\\\",",
      "    \"GlobalIPv6Address\\": \\\"\\\",",
      "    \"GlobalIPv6PrefixLen\\": 0,",
      "    \"MacAddress\\": \\\"02:42:ac:11:00:02\\\",",
      "    \"DriverOpts\\": null,",
      "  },",
      "},",
      "},",
      "},",
      "}"
    ]
  }
}

PLAY RECAP *****
192.168.225.27      : ok=11   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Server is accessible to world:



Ansible is only used for configuration purpose and it is python library

Task 1 is Successfully completed