Subhakanta Mishra
Subhamishra.in@gmail.com

Configuration Management with Ansible and Terraform

Course-End Project: Ansible playbook for deploying html code and configurations on NGINX server

Objective

To automate the provisioning of infrastructure and configuring it with integration of Terraform and Ansible.

Real-time scenario:

Royal Hotel is a globally leading chain of hotels. Recently, as part of scaling up operations, they aim to automate every operation in the hotel. For this, multiple applications are onboarded within all the hotel's main server. To keep these applications up and running and to scale them appropriately, they need fully managed virtual machines on AWS. They want to have an automated provisioned infrastructure through which they can create a new developer VM and manage some developer configurations on that server.

Tasks

The following tasks outline the process of provisioning and infrastructure using Terraform and parameterised Jenkins job:

1. Configure Terraform with new ssh key which will be used as key pair for launching VMs.

2. Configure AWS CLI with access key and secret key to establish connection remotely

3. Write Terraform script to provision and empty sandbox.

4. Add various setting to the sandbox like VPC, security group, route table, subnets, and key pair.

5. Create Ansible playbook which will be invoked by Terraform for configuration management operations.

6. Executing Terraform script with the created keys to establish connection and configure the provisioned VM.

Subhakanta Mishra
Subhamishra.in@gmail.com

Solution

Step1: Create Inventory file to store worker details

cd

ssh-keygen

cat /root/.ssh/id_rsa.pub

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:0NpNWvrNupqluztitViWoxIPadqFDq6362NLdOcswaw root@ip-172-31-22-14
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|       .         |
|      . . o      |
|   o   + *       |
|  . =oo S..      |
| ..o*=. *. o     |
| .E*.=o* oo o    |
| .* +.* o+ .     |
|.=== o .B*o.     |
+----[SHA256]-----+
root@ip-172-31-22-14:~# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDEUdozTyuVFLGfF2Y/DIGEdAm7kNn9UBKk9SicTrfuEaiEYJ5/c0gu+uMDwLNvHTWAh0XV7Xhr1Vc2u5dHIrijPRh2vCoH3n5iId+d
hTniXhZwFJAld7roBasvoRaQw6QHWpJRwyLDJyxXfRyqswMD7TVe3rP4uPgkRbl6qOXZKDpPPSqTcd7lW3j8FJSf2/wxUf4iSNEQFg8RXiSJJ9JuFdJrVpMOZ5RXh1qvgqHe5e4igd4l
P3MjnfjZGJypv07GVdeQ4/wBcl7JdWpox16WVy5LaNIraX5ZaenDdypUc9WmLLjEKiGcRXBLUrTcd8C5aoylyQm152GnpjRuznLEfxIze+Ozze+yT/h5JATBxxZkzs1tcapzCarX12eY
1OU10XASW/0f86LYq3cn4DBuv/skADosXkJRQffniHYepAvzneyME8Q0/T6QKvt6x2BUQRkeB+2PgDrzEIYW9tq+AIkDrQvbtDupW72s6GRj2pBIbQfIcMKdZsyZHVCBsW0= root@ip
-172-31-22-14
```

Step-2: Configure AWS.

```
root@ip-172-31-22-14:~# aws configure
AWS Access Key ID [None]: AKIAX77G2QBQTC26HQKV
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
root@ip-172-31-22-14:~#
```

Step-3: Create worker node in AWS EC2 and Copy the public key in .ssh folder

```
root@ip-172-31-80-15:~/.ssh# echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDEUdozTyuVFLGfF2Y/DIGEdAm7kNn9UBKk9SicTrfuEaiEYJ5/c0gu+uMDwLNvHTWAh0XV7Xhr1Vc2u5dHIrijPRh2vCoH3n5iId
+dhTniXhZwFJAld7roBasvoRaQw6QHWpJRwyLDJyxXfRyqswMD7TVe3rP4uPgkRbl6qOXZKDpPPSqTcd7lW3j8FJSf2/wxUf4iSNEQFg8RXiSJJ9JuFdJrVpMOZ5RXh1qvgqHe5e4igd4lP3MjnfjZGJypv07GVdeQ4/wBcl7JdWp
ox16WVy5LaNIraX5ZaenDdypUc9WmLLjEKiGcRXBLUrTcd8C5aoylyQm152GnpjRuznLEfxIze+Ozze+yT/h5JATBxxZkzs1tcapzCarX12eY1OU10XASW/0f86LYq3cn4DBuv/skADosXkJRQffniHYepAvzneyME8Q0/T6QKvt6
x2BUQRkeB+2PgDrzEIYW9tq+AIkDrQvbtDupW72s6GRj2pBIbQfIcMKdZsyZHVCBsW0= root@ip-172-31-22-14" >> ~/.ssh/authorized_keys
root@ip-172-31-80-15:~/.ssh#
```

Step-4: Create the inventory file and ansible.cfg file and test the connection.

vim myinventory

Subhakanta Mishra
Subhamishra.in@gmail.com

[webserver]

<public ip of worker ec2 instance>

Save the file

```
root@ip-172-31-19-44:~# vim myinventory
root@ip-172-31-19-44:~# cat myinventory
[webserver]
18.207.215.173
```

 # pwd

Copy the path of the directory

 # vim ansible.cfg

[defaults]

inventory = /root/myinventory

Save the file

```
root@ip-172-31-22-14:~# vim myinventory
root@ip-172-31-22-14:~# ls
myinventory  snap
root@ip-172-31-22-14:~# vim ansible.cfg
root@ip-172-31-22-14:~# cat ansible.cfg
[defaults]
inventory = /root/myinventory

root@ip-172-31-22-14:~#
```

Run ping command to check the connection

Validate the setup :

ansible webserver -m ping

```
root@ip-172-31-22-14:~# ansible webserver -m ping
52.70.138.184 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
root@ip-172-31-22-14:~#
```

Subhakanta Mishra
Subhamishra.in@gmail.com

Step 5: Write a YAML playbook with tasks for installing Nginx

vim playbook.yml

```
- name: Deploying project 1
 hosts: webserver
 become: true
 vars:
  pkg_name: nginx
 tasks:
 - name: update the apt repo
   command: apt-get update
 - name: Install nginx application
   package: name={{ pkg_name }} state=present
```

```
File  Edit  View  Search  Terminal  Help
- name: Deploying project 2
  hosts: webserver
  become: true
  vars:
   pkg_name: nginx
  tasks:
  - name: update the apt repo
    command: apt-get update
  - name: Install nginx application
    package: name={{ pkg_name }} state=present
```

Save and run the playbook

```
root@ip-172-31-22-14:~# ansible-playbook playbook.yml

PLAY [Deploying project 2] ***************************************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [52.70.138.184]

TASK [update the apt repo] ***************************************************************************************
changed: [52.70.138.184]

TASK [Install nginx application] *********************************************************************************
changed: [52.70.138.184]

PLAY RECAP *******************************************************************************************************
52.70.138.184              : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ip-172-31-22-14:~#
```

Subhakanta Mishra
Subhamishra.in@gmail.com

Step 6: Update the nginx configuration using Jinja2 template

cd

cp /etc/nginx/sites-available/default .

mv default default.j2

vim default.j2

Update port 80 with a variable {{ http_port }} , as shown below

```
#
# This file will automatically load configuration files
# applications, such as Drupal or Wordpress. These appli
# available underneath a path with that package name, su
#
# Please see /usr/share/doc/nginx-doc/examples/ for more
##

# Default server configuration
#
server {
        listen {{ http_port }} default_server;
        listen [::]:{{ http_port }} default_server;

        # SSL configuration
```

Step-7: Update the playbook with jina2 template tasks

vim playbook.yml

```
- name: Deploying project 1
  hosts: webserver
  become: true
  vars:
   pkg_name: nginx
   http_port: 8282
  tasks:
```

Subhakanta Mishra
Subhamishra.in@gmail.com

- name: update the apt repo

  command: apt-get update

- name: Install nginx application

  package: name={{ pkg_name }} state=present

- name: Update the ports in nginx config file

  template: src=default.j2 dest=/etc/nginx/sites-available/default

  notify: Restart nginx service

handlers:

- name: Restart nginx service

  service: name={{ pkg_name }} state=restarted

```
File  Edit  View  Search  Terminal  Help
- name: Deploying project 2
  hosts: webserver
  become: true
  vars:
   pkg_name: nginx
   http_port: 8282
  tasks:
  - name: update the apt repo
    command: apt-get update
  - name: Install nginx application
    package: name={{ pkg_name }} state=present
  - name: Update the ports in nginx config file
    template: src=default.j2 dest=/etc/nginx/sites-available/default
    notify: Restart nginx service
  handlers:
  - name: Restart nginx service
    service: name={{ pkg_name }} state=restarted
```

Save the playbook and run it

Subhakanta Mishra
Subhamishra.in@gmail.com

```
root@ip-172-31-22-14:~# ansible-playbook playbook.yml

PLAY [Deploying project 2] ***************************************************************************************

TASK [Gathering Facts] *******************************************************************************************
ok: [52.70.138.184]

TASK [update the apt repo] ***************************************************************************************
changed: [52.70.138.184]

TASK [Install nginx application] *********************************************************************************
ok: [52.70.138.184]

TASK [Update the ports in nginx config file] *********************************************************************
changed: [52.70.138.184]

RUNNING HANDLER [Restart nginx service] **************************************************************************
changed: [52.70.138.184]

PLAY RECAP *******************************************************************************************************
52.70.138.184              : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ip-172-31-22-14:~#
```

Step-8: Create an HTML code and deploy it on nginx sever

vim index.html

<marquee> Capstone project - CMAT </marquee>

<h1> This is project 2 </h1>

<h2> Created by Subhakant </h2>

<marquee> Look my project got executed successfully </marquee>

```
<marquee> Capstone project - CMAT </marquee>
<h1> This is project 2 </h1>
<h2> Created by Subhakant </h2>
<marquee> Look my project got executed successfully </marquee>
```

Save the file

Update the playbook with the new tasks

vim playbook.yml

- name: Deploying project 1

  hosts: webserver

  become: true

  vars:

   pkg_name: nginx

   http_port: 8282

  tasks:

Subhakanta Mishra
Subhamishra.in@gmail.com

- name: update the apt repo

  command: apt-get update

- name: Install nginx application

  package: name={{ pkg_name }} state=present

- name: Update the ports in nginx config file

  template: src=default.j2 dest=/etc/nginx/sites-available/default

  notify: Restart nginx service

- name: Deploy HTML code on nginx server

  copy: src=index.html dest=/var/www/html

  notify: Restart nginx service

handlers:

- name: Restart nginx service

  service: name={{ pkg_name }} state=restarted

```
- name: Deploying project 2
  hosts: webserver
  become: true
  vars:
   pkg_name: nginx
   http_port: 8282
  tasks:
  - name: update the apt repo
    command: apt-get update
  - name: Install nginx application
    package: name={{ pkg_name }} state=present
  - name: Update the ports in nginx config file
    template: src=default.j2 dest=/etc/nginx/sites-available/default
    notify: Restart nginx service
  - name: Deploy HTML code on nginx server
    copy: src=index.html dest=/var/www/html
    notify: Restart nginx service

  handlers:
  - name: Restart nginx service
    service: name={{ pkg_name }} state=restarted
```

Save the file and execute the playbook.

Subhakanta Mishra
Subhamishra.in@gmail.com

```
root@ip-172-31-22-14:~# ansible-playbook playbook.yml

PLAY [Deploying project 2] ***************************************************************************************************

TASK [Gathering Facts] *******************************************************************************************************
ok: [52.70.138.184]

TASK [update the apt repo] ***************************************************************************************************
changed: [52.70.138.184]

TASK [Install nginx application] *********************************************************************************************
ok: [52.70.138.184]

TASK [Update the ports in nginx config file] *********************************************************************************
ok: [52.70.138.184]

TASK [Deploy HTML code on nginx server] **************************************************************************************
changed: [52.70.138.184]

RUNNING HANDLER [Restart nginx service] **************************************************************************************
changed: [52.70.138.184]

PLAY RECAP *******************************************************************************************************************
52.70.138.184              : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ip-172-31-22-14:~#
```

Go to the browser, take the public ip of worker node

Publicip:8282