PG DO - CONFIGURATION MANAGEMENT WITH ANSIBLE AND TERRAFORM

DEPLOYING WEB APPLICATION USING ANSIBLE

Git Repository:
https://github.com/sravan1990/Simplilearn_Config_mgmt_using_ansible_and_terraform.git

PRESENTED BY: VENKATA SRAVAN KUMAR CHIVUKULA

.
.

DEPLOYING WEB APPLICATION USING ANSIBLE

PROJECT AGENDA

To create an automation script to deploy an application using Ansible.

SCENARIO

You have joined as a DevOps engineer in XYZ Pvt. Ltd. It is a platform where individuals can create their profile and start blogging on various topics. The application is ready to be hosted on a server. You are tasked with implementing an Ansible script to deploy this application on a remote Nginx server.

INDUSTRY RELEVANCE

The following tools used in this project serve specific purposes within the industry:
1. Ansible: Ansible automates IT tasks, streamlining configuration management, application deployment, and orchestration. It uses simple, human-readable YAML files called playbooks, allowing easy setup and management of complex IT environments

Tools required: Ansible
Terraform,
AWS account with security credentials,
Keypair

Expected Deliverables: Launch an EC2 instance using Terraform
Connect to the instance
Execute the playbook to deploy the web application on the remote server

SOLUTION EXECUTION STEPS

1. LAUNCH AN EC2 INSTANCE USING TERRAFORM REPRESENTING A REMOTE SERVER

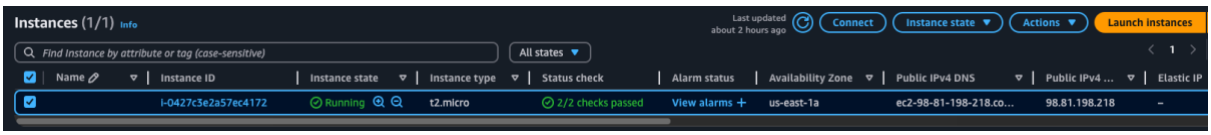
a. Create a **main.tf** file. (Check the repository)

b. initialize terraform using command :
$ terraform init

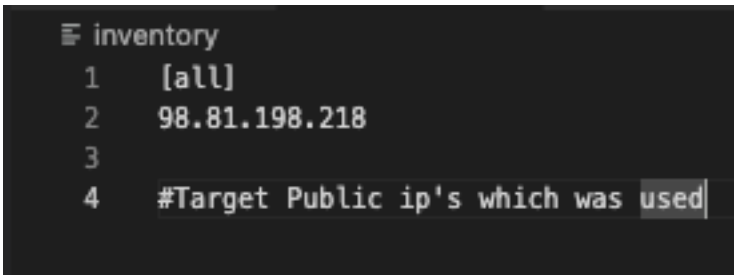
c. Run plan :
$ terraform plan

d. run apply :
$ terraform apply

Now check whether instance is created in AWS.



2. CREATE AN INVENTORY FILE TO DEFINE THE REMOTE SERVER



3. CREATE A YAML PLAYBOOK WITH TASKS FOR INSTALLING NGINX, COPYING WEB APPLICATION FILES, DEPLOYING THE NGINX CONFIGURATION, AND ENABLING THE SITE.

a. Create a **ngnix.yml** file. (Check the git repository)

4. CREATE A DIRECTORY FOR TEMPLATES AND AN ANSIBLE PLAYBOOK FOR THE NGINX CONFIGURATION

a. Templates can be saved under a dedicated directory on the remote

var/www/html/application/templates

b. Ansible uses a specific type of template called Jinja Templates (.j2)

c. For the ansible code check the git repository for **ngnix.yml** file.

5. DEFINE VARIABLES IN THE PLAYBOOK FOR APPLICATION DETAILS AND NGINX CONFIGURATION

a. Check the git repository for **ngnix.yml** file.

6. INCLUDE TASKS IN THE PLAYBOOK FOR INSTALLING NGINX, COPYING APPLICATION FILES, DEPLOYING NGINX CONFIGURATION, AND ENABLING THE NGINX SITE

a. Check the git repository for **ngnix.yml** file.

7. EXECUTE THE PLAYBOOK TO DEPLOY THE WEB APPLICATION ON THE REMOTE SERVER.

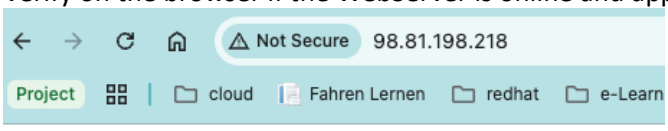
- Run playbook
  % ansible-playbook -i inventory nginx.yml

```
sravankumar@MacBookAir project5 % ansible-playbook -i inventory nginx.yml

PLAY [all] ********************************************************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************************************
ok: [98.81.198.218]

TASK [authorized_key] *********************************************************************************************************************************************
ok: [98.81.198.218]

TASK [Install nginx] **********************************************************************************************************************************************
changed: [98.81.198.218]

TASK [Create directory for static content] *************************************************************************************************************************
changed: [98.81.198.218]

TASK [Create "index.html" file with app content] ******************************************************************************************************************
changed: [98.81.198.218]

TASK [Copy "index.html" to default Nginx location] ****************************************************************************************************************
changed: [98.81.198.218]

TASK [Enable default Nginx website] *******************************************************************************************************************************
ok: [98.81.198.218]

TASK [Restart Nginx] **********************************************************************************************************************************************
changed: [98.81.198.218]

PLAY RECAP ********************************************************************************************************************************************************
98.81.198.218              : ok=8    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

sravankumar@MacBookAir project5 %
```

Verify Nginx server installation and restart

```
ubuntu@ip-172-31-85-108:~$ systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-01-12 13:43:53 UTC; 5min ago
       Docs: man:nginx(8)
    Process: 6143 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 6145 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 6146 (nginx)
      Tasks: 2 (limit: 1130)
     Memory: 1.7M (peak: 2.0M)
        CPU: 8ms
     CGroup: /system.slice/nginx.service
             ├─6146 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─6147 "nginx: worker process"

Jan 12 13:43:53 ip-172-31-85-108 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Jan 12 13:43:53 ip-172-31-85-108 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-85-108:~$
```

Verify on the browser if the WebServer is online and application runs ☺



Sammy The Shark

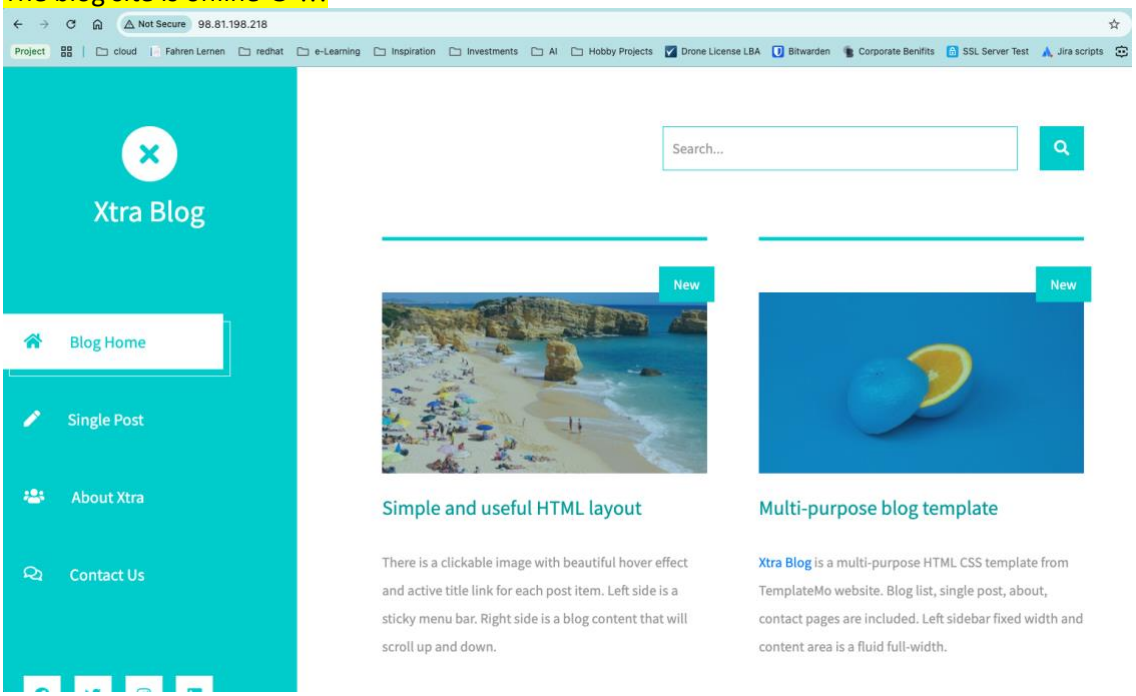
About this site

8. RERUN WITH BETTER WEBSERVER TEMPLATE AND MODIFIED CONFIG

☺ Rerun Playbook with a better website template and a dedicated directory for Templates(Sample Jinja template copied) ☺

```
sravankumar@MacBookAir project5 % ansible-playbook -i inventory nginx.yml

PLAY [all] ********************************************************************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************************************************
ok: [98.81.198.218]

TASK [authorized_key] *********************************************************************************************************************************************************
ok: [98.81.198.218]

TASK [Update apt cache and Install nginx] *************************************************************************************************************************************
ok: [98.81.198.218]

TASK [Create directory for static content and templates] ***********************************************************************************************************************
changed: [98.81.198.218]

TASK [Copy templates to the respective directory on remote server] ************************************************************************************************************
changed: [98.81.198.218]

TASK [Copy website files to server's document root] ***************************************************************************************************************************
ok: [98.81.198.218]

TASK [Copy website files to default Nginx server location] ********************************************************************************************************************
changed: [98.81.198.218]

TASK [Enable default Nginx website] *******************************************************************************************************************************************
ok: [98.81.198.218]

TASK [Restart Nginx] **********************************************************************************************************************************************************
changed: [98.81.198.218]

TASK [Allow all access to tcp port 80] ****************************************************************************************************************************************
ok: [98.81.198.218]

PLAY RECAP ********************************************************************************************************************************************************************
98.81.198.218              : ok=10   changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

sravankumar@MacBookAir project5 %
```

The blog site is online ☺ !!!



CONCLUSION / RESULT

Successfully ran an ansible playbook to

- connect to a remote EC2 instance
- Configure and Install Nginx server on the remote
- Copy the templates and website files to remote
- Enable Nginx server
- Update and run application config successfully!!