Assignment: - by Shamik Guha Ray

Successfully deployed the necessary infrastructure setup to execute below assignment:

Problem/question statement

Design an automation deployed with GITHUB/GITLAB pipelines to create load balanced simple nginx hosted on 1 or more virtual machines on AWS or Azure with the following assumption :

1. CIDR retrieve from REST API https://FDQN/vend_ip return
```
{
  "ip_address":"192.168.0.0",
  "subnet_size":"/16"
}
```
2. Create subnets with size /24
3. Generate SSH key for VM credential
4. Take into consideration CSP Best Practices such as security and resiliency
5. Take into consideration coding/scripting practices
6. Leverage on native cloud metrics/logging for error handling
7. Can use bash/terraform/python/powershell for the stack, github or github for the IAC pipeline

Provide evidence of the successful execution

Deployment Steps:

Repository Creation:

- Created a new repository named "github-nginx-terraform-aws".
- GitHub Repo Link is attached here.

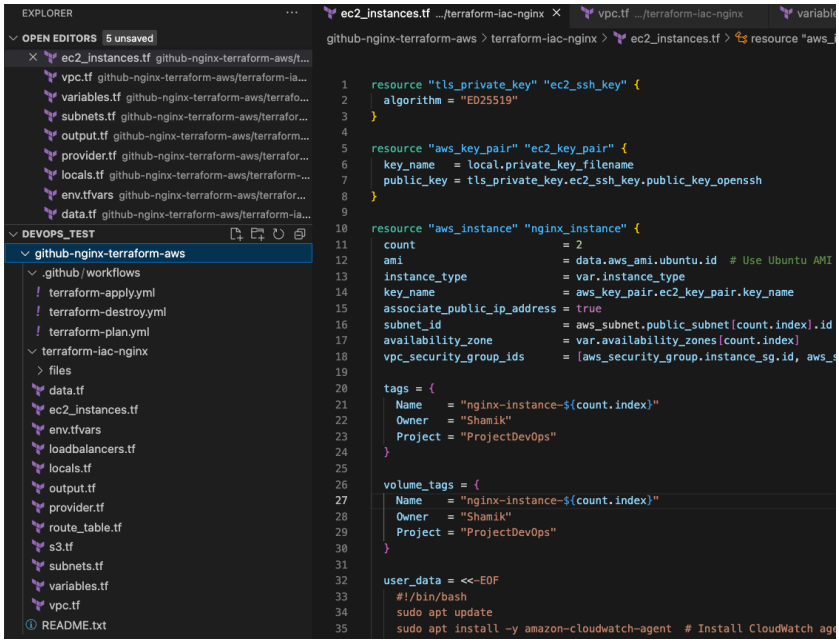
Local Setup:

- Cloned the repository locally.
- Created Terraform configurations.
- Pushed the configurations to the master branch of the repository. The GitHub Actions workflows got initiated upon “git push to master”.

```
sguharay@s-guharay-n60cm github-nginx-terraform-aws % git add .
sguharay@s-guharay-n60cm github-nginx-terraform-aws % git commit -m "Deploy nginx server infra to AWS"
[master a68d37c] Deploy nginx server infra to AWS
 1 file changed, 4 insertions(+), 4 deletions(-)
sguharay@s-guharay-n60cm github-nginx-terraform-aws %
sguharay@s-guharay-n60cm github-nginx-terraform-aws %
sguharay@s-guharay-n60cm github-nginx-terraform-aws % git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 10 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 408 bytes | 408.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:ShamikGray/github-nginx-terraform-aws.git
   e021051..a68d37c  master -> master
sguharay@s-guharay-n60cm github-nginx-terraform-aws %
```

Terraform Configuration:

- Utilized VS Code to craft the Terraform configuration files.

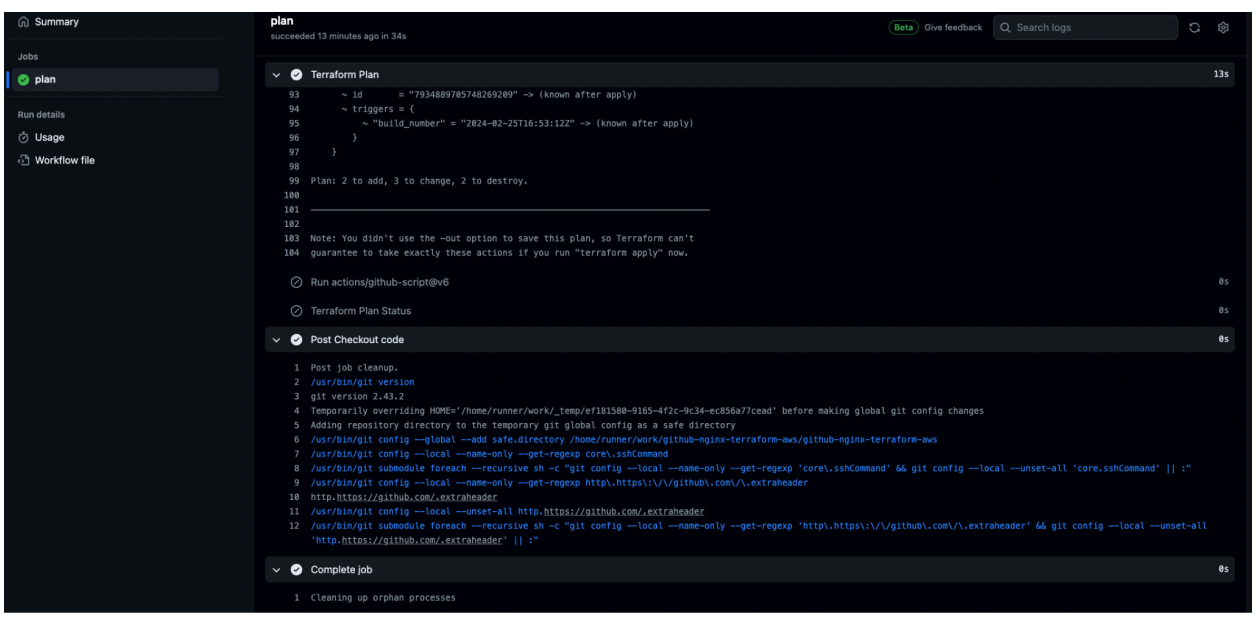


Workflow Utilization:

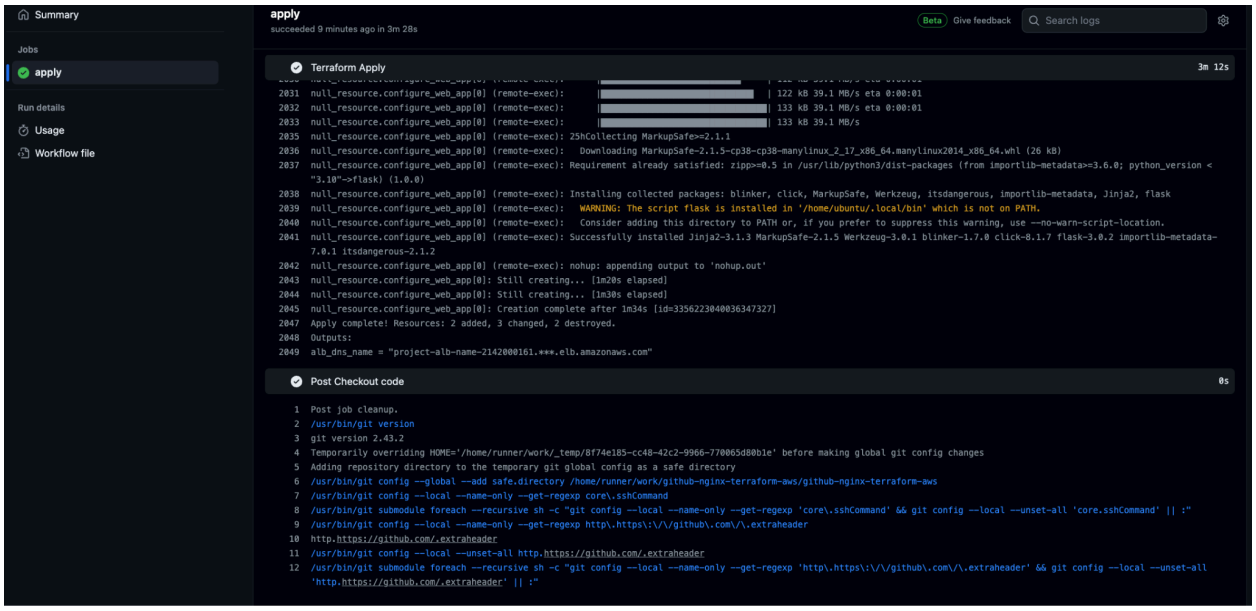
- Implemented two separate workflows: Terraform Plan and Terraform Apply.

By following these steps, the necessary infrastructure for the assignment was successfully deployed. Below is the screenshot of Github Actions Deployment:

Terraform Plan - GitHub Actions Workflow:



Terraform Apply - GitHub Action Workflow:



Post-Deployment Validation:

Infrastructure Verification:

- Ensured the correct provisioning of all infrastructure resources post-deployment.

Accessibility Testing:

- Tested accessibility to nginx servers using the load balancer's DNS name.

Multi-AZ Availability Testing:

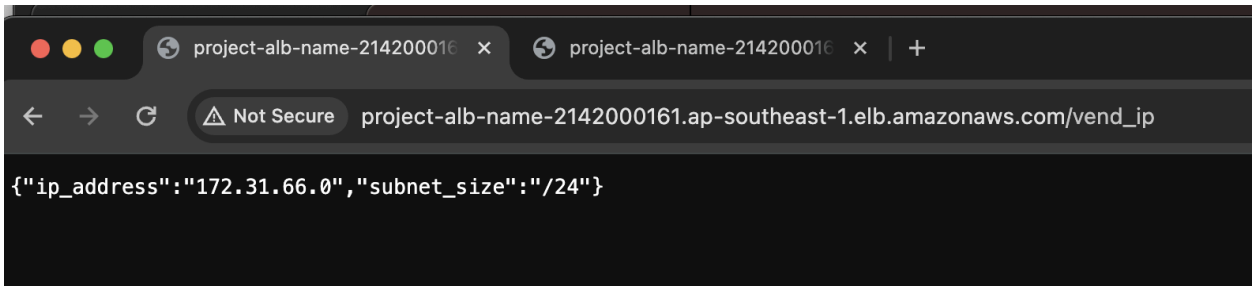
- Validated the load balancer's ability to distribute traffic across instances in multiple availability zones.

Steps Taken:

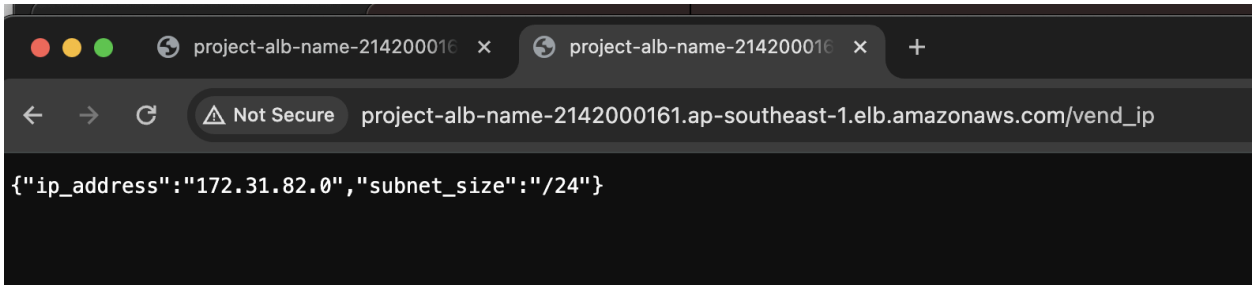
Accessed the following DNS resolver to perform post-deployment validation:
"http://project-alb-name-2142000161.ap-southeast-1.elb.amazonaws.com/vend_ip"

Screenshot of the Content Served by Our Two Nginx EC2 Servers:

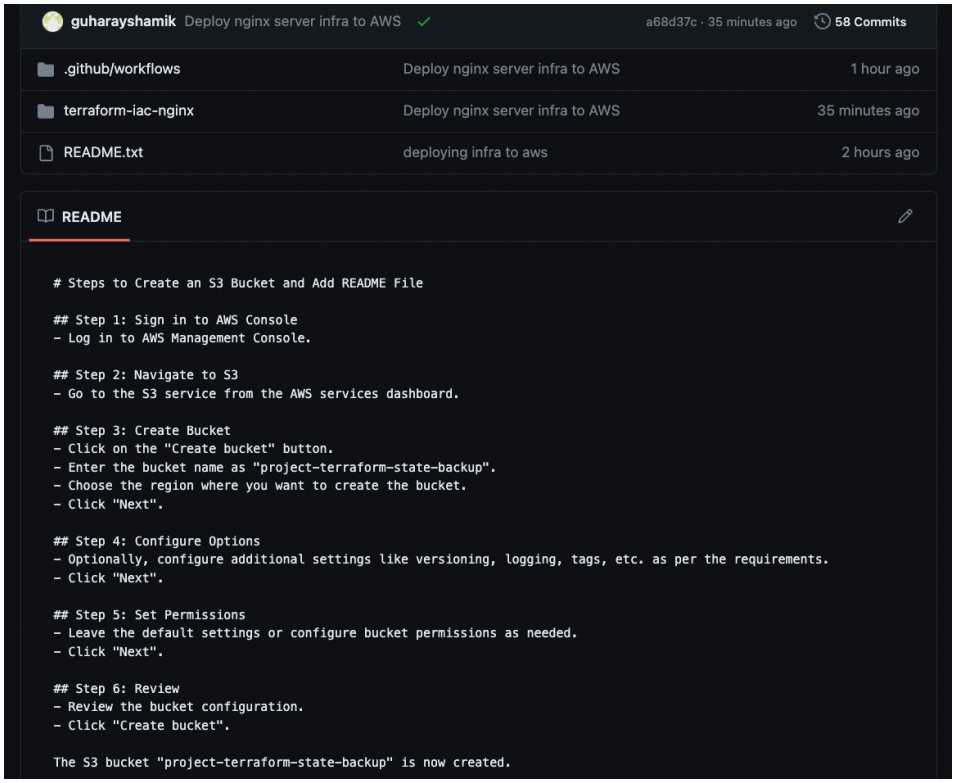
1)



2)



Attached the README containing the steps to setup the s3 bucket before triggering the terraform init/plan/apply.



Please find the screenshots of AWS resources deployed using the workflow: