Week 5- Group A (Task)

Two Tier Infrastructure Setup with Packer and Terraform

Task

Fork/Copy: • application code repo: https://github.com/techbleat/fruits-veg_market

• Infrastructure code repo: https://github.com/techbleat/class2025iac

Redo Week 4 task, this time creating 2 AMIs using Packer; Node 1 [NGINX] and Node 2 [Java 17 /Python3]. Then, use Terraform to spin up the following infrastructure leveraging the AMIs created.

• Node 1: A frontend/Tier 1 system running NGINX

• Node 2: A backend/Tier 2 system running Python 3 o should run on port 8080

• Node 3: A backend/Tier 2 system running Java 17 o should run on port 9090 After completion:

1. share screenshots of your infrastructure

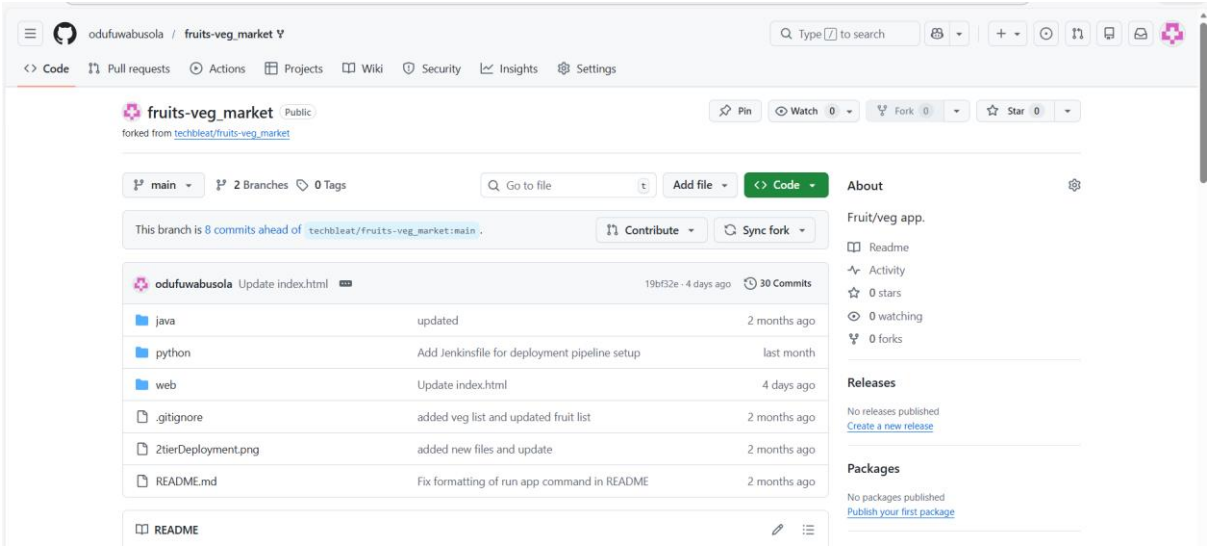
2. Share your GitHub repo(s) Remember to exclude .terraform directory and other files not needed when pushing your code to the repo.

Implementation Steps

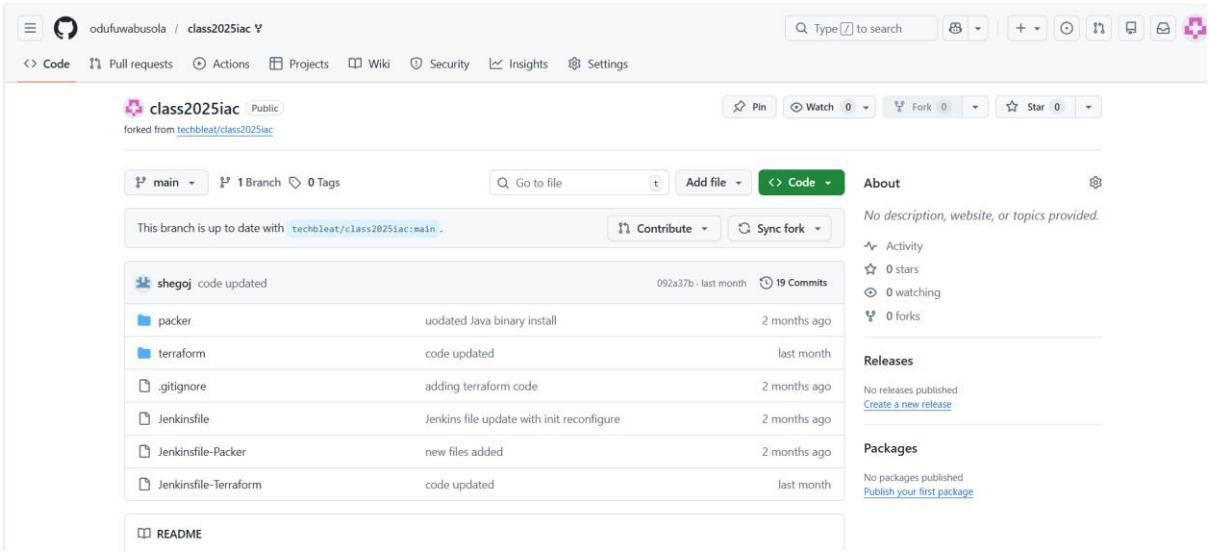
1. Fork/copy the application code repo and the infrastructure code repo
2. Created AMI images using packer
3. Created a tool server for execution
4. Created terraform to spin up the following infrastructure (nginx, python (8080) and java (9090)
5. Configure the load balancing for frontend/Tier 1 – Nginx, backend/Tier2- Python on port 8080 and backend/Tier 2 java on port 9090.

Fork/Copy

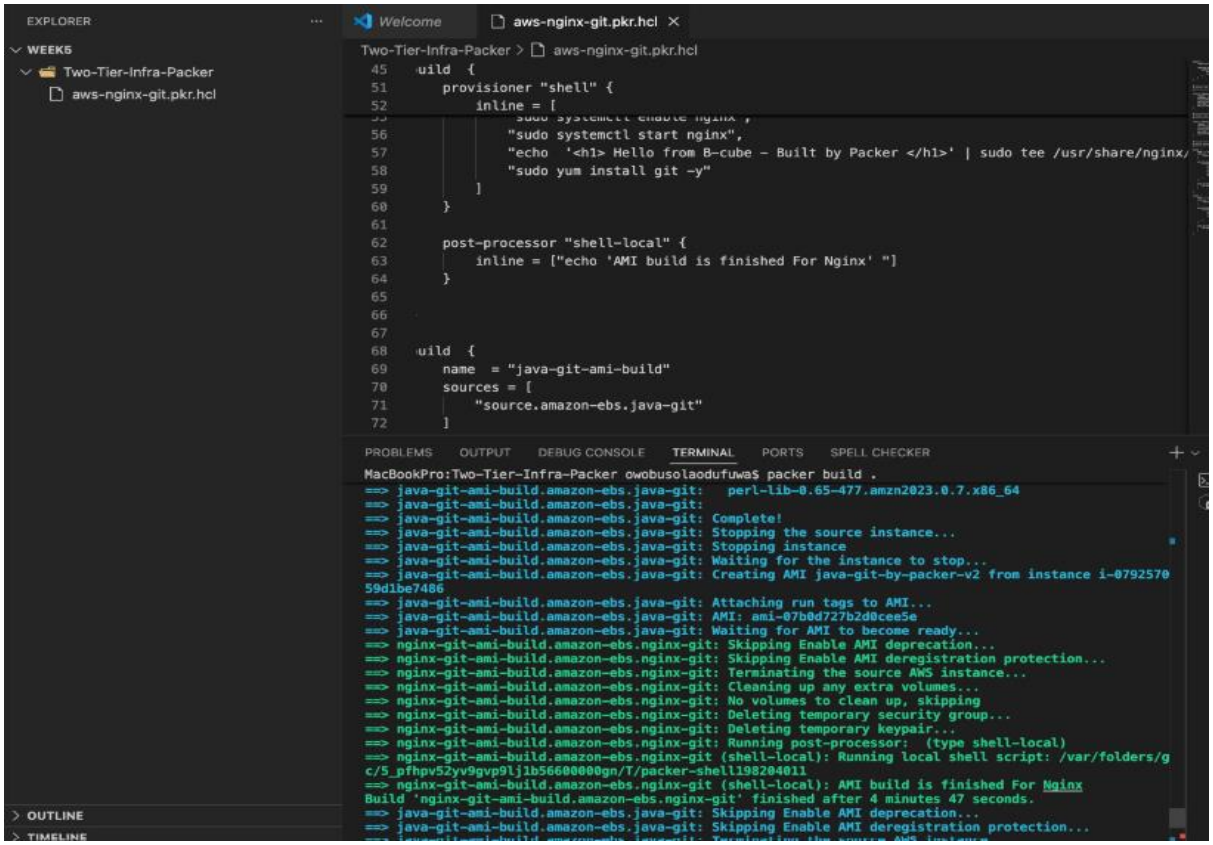
• application code repo: https://github.com/techbleat/fruits-veg_market



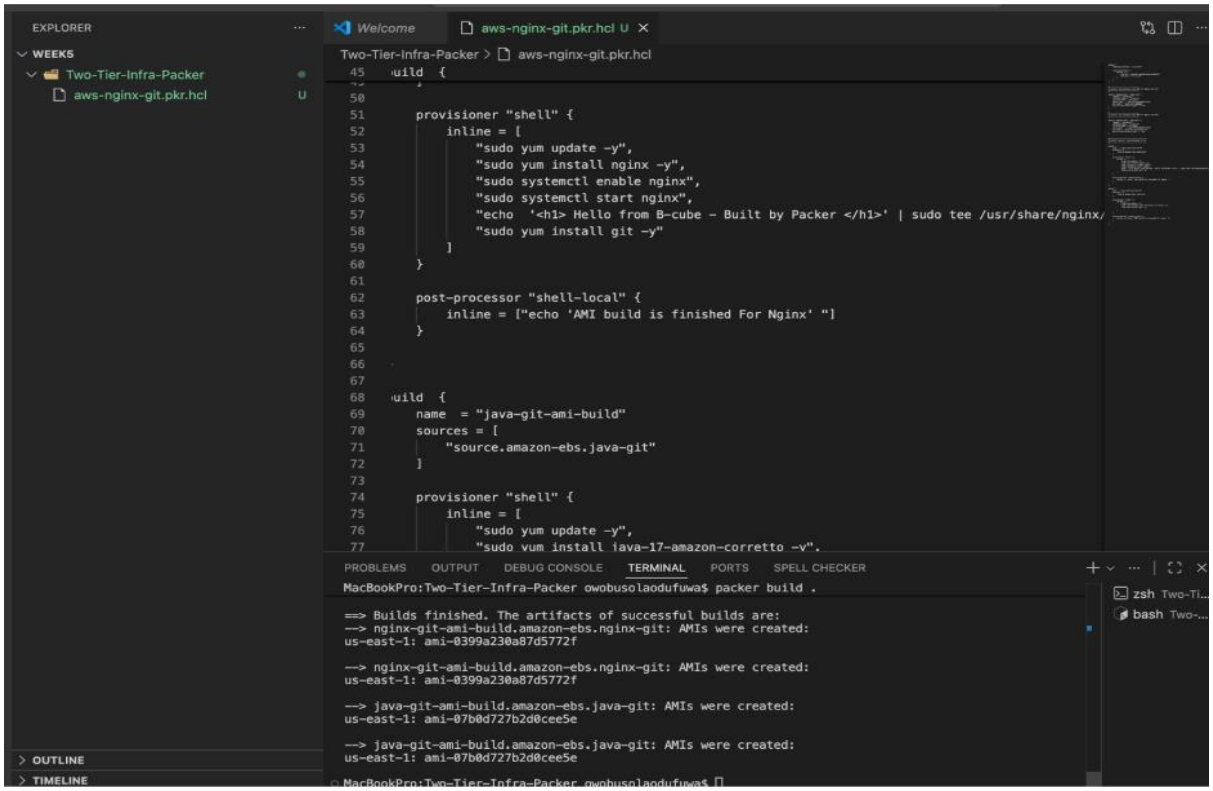
• Infrastructure code repo: https://github.com/techbleat/class2025iac



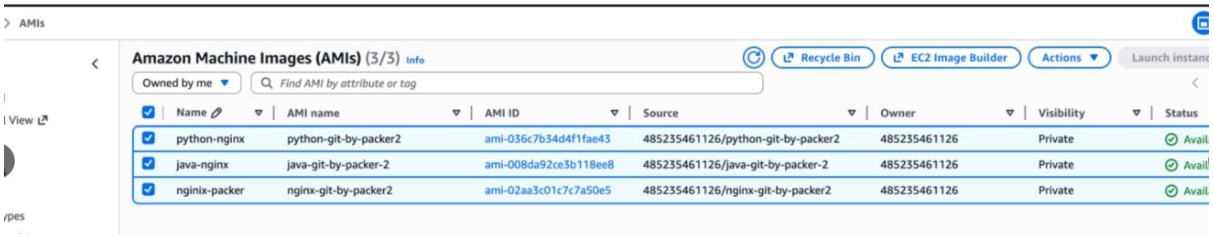
Packer



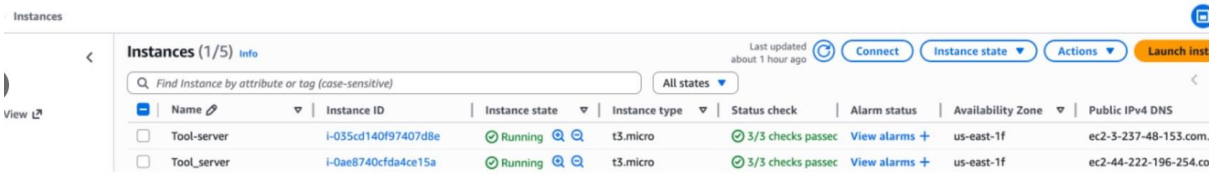
Packer successfully Installed



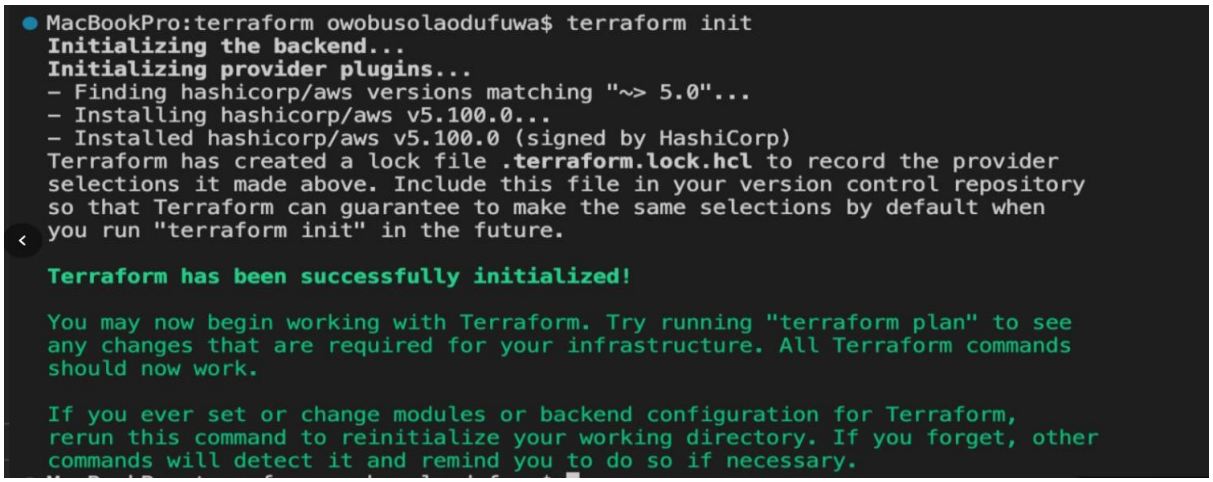
AMI Images created from Packer



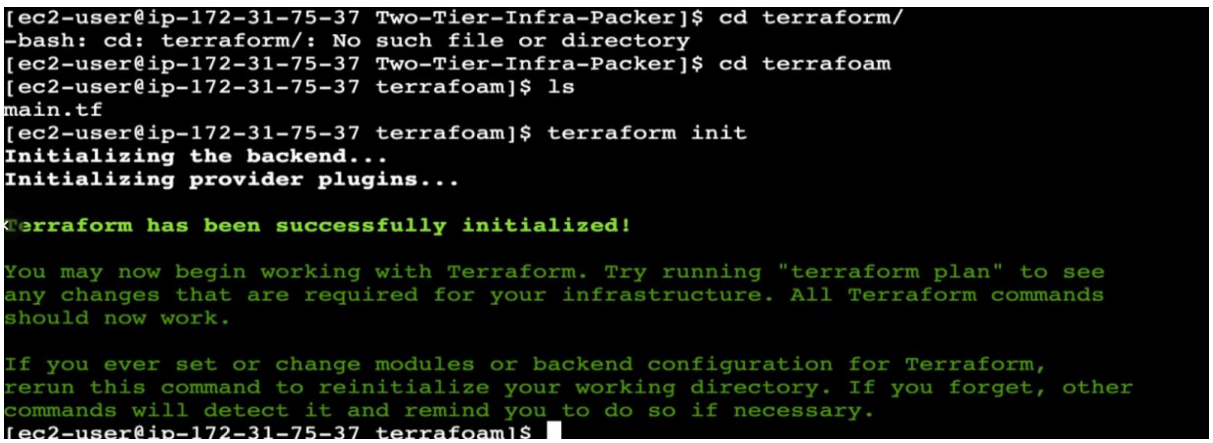
Tool Server



Terraform initiated (vscode)

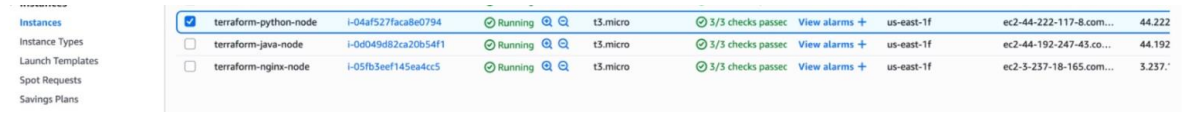


Terrafoam initiated (aws- ec2)

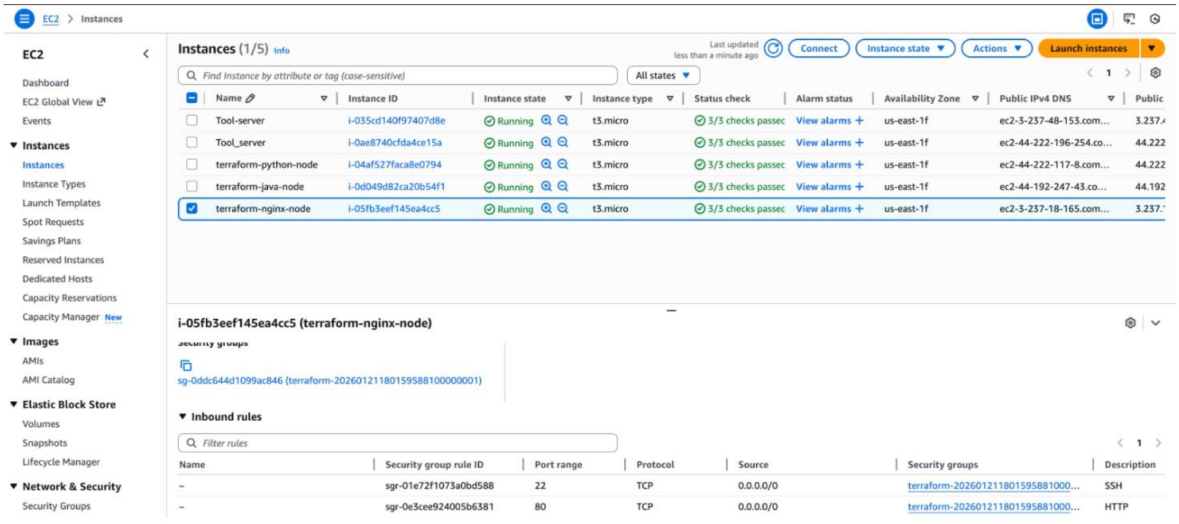


```
Plan: 6 to add, 0 to change, 3 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_security_group.java_web: Destroying... [id=sg-031e80f167d090ae5]
aws_security_group.nginx_web: Destroying... [id=sg-09edebb86d4ed436b]
aws_security_group.python_web: Destroying... [id=sg-0e2f3130fe454e390]
aws_security_group.nginx_web: Destruction complete after 1s
aws_security_group.nginx_web: Creating...
aws_security_group.python_web: Destruction complete after 1s
aws_security_group.python_web: Creating...
aws_security_group.java_web: Destruction complete after 1s
aws_security_group.java_web: Creating...
aws_security_group.python_web: Creation complete after 2s [id=sg-01c8815caa4d4e7e7]
aws_security_group.nginx_web: Creation complete after 2s [id=sg-0ddc644d1099ac846]
aws_security_group.java_web: Creation complete after 2s [id=sg-0e3bca4d7e7c40e8d]
aws_instance.python-node: Creating...
aws_instance.nginx-node: Creating...
aws_instance.java-node: Creating...
aws_instance.python-node: Still creating... [00m10s elapsed]
aws_instance.nginx-node: Still creating... [00m10s elapsed]
aws_instance.java-node: Still creating... [00m10s elapsed]
aws_instance.nginx-node: Creation complete after 13s [id=i-05fb3eef145ea4cc5]
aws_instance.java-node: Creation complete after 13s [id=i-0d049d82ca20b54f1]
aws_instance.python-node: Creation complete after 13s [id=i-04af527faca8e0794]

Apply complete! Resources: 6 added, 0 changed, 3 destroyed.
[ec2-user@ip-172-31-75-37 terraform]$
```

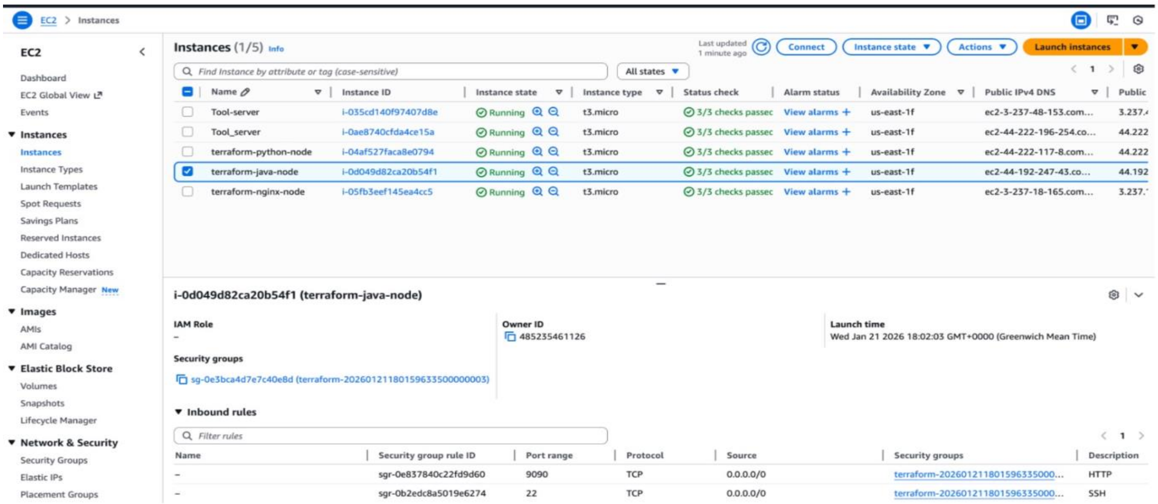
The three nodes were created using terraform



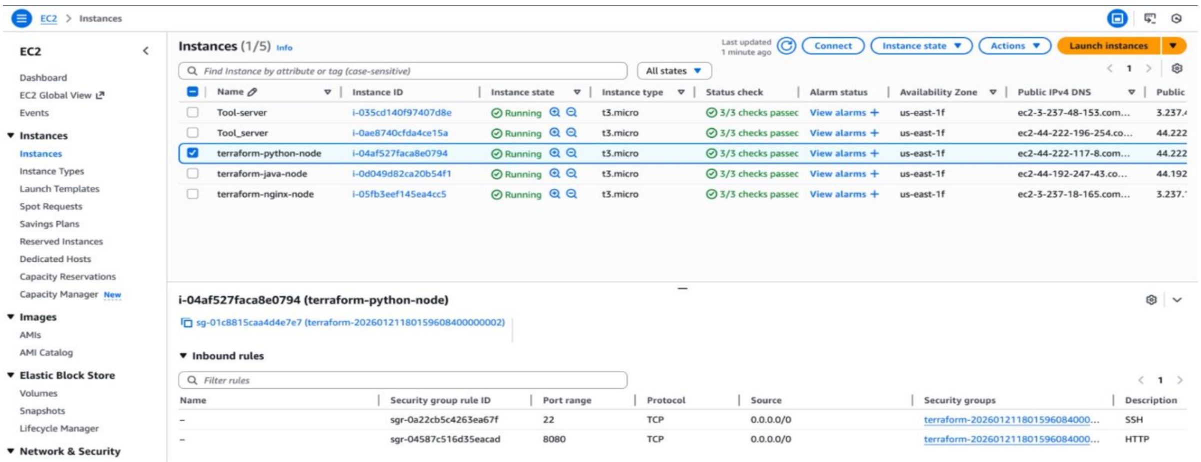
Nginx from Terraform



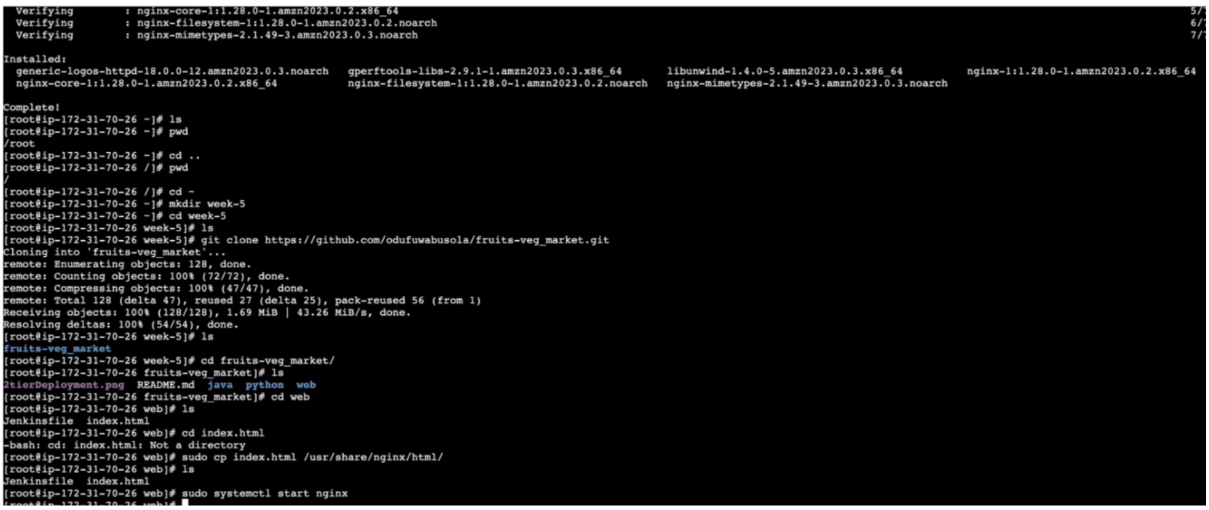
Java from Terraform running on port 9090



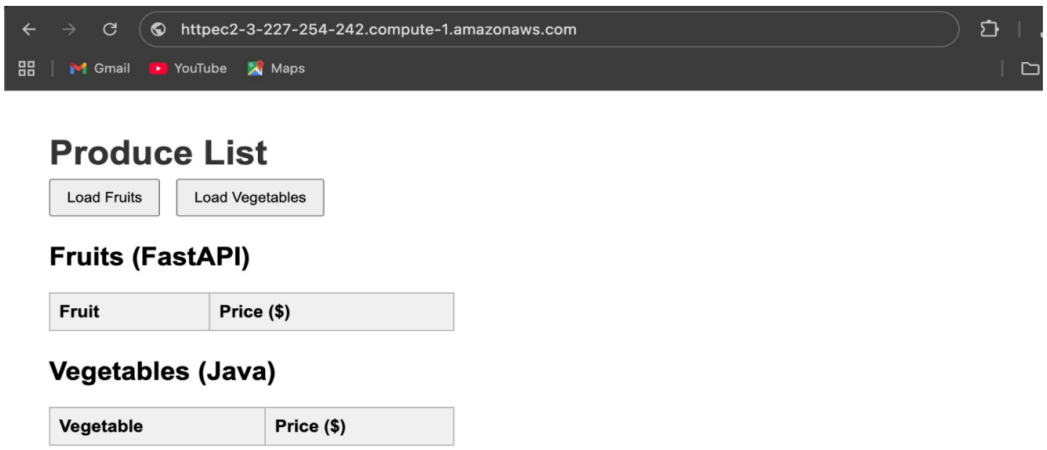
Python from Terraform running on port 8080



Nginx

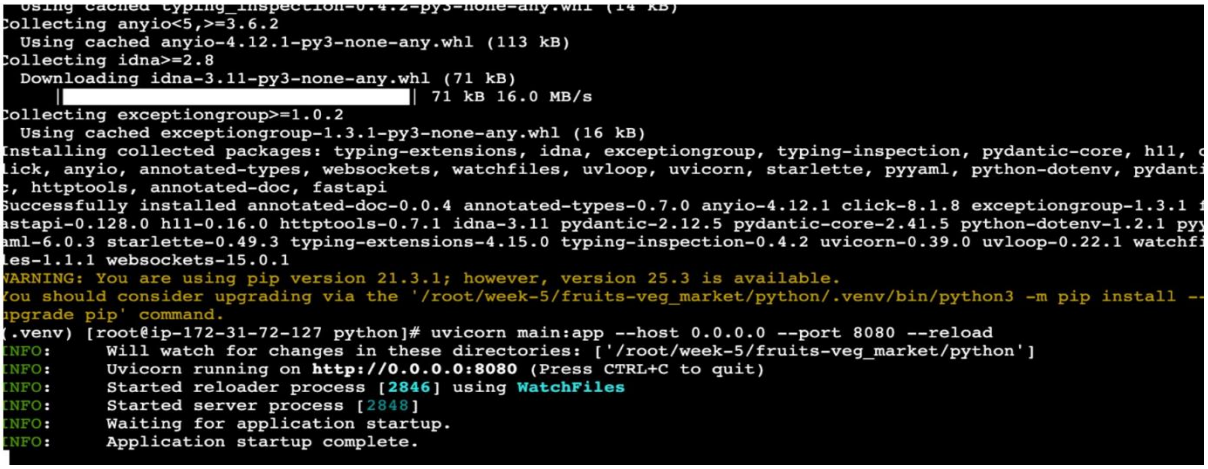


Frontend Load Balancer



Python Load balancer

Python running on port 8080



Java Load Balancer

Produce List

Load Fruits | Load Vegetables

Loaded ✓

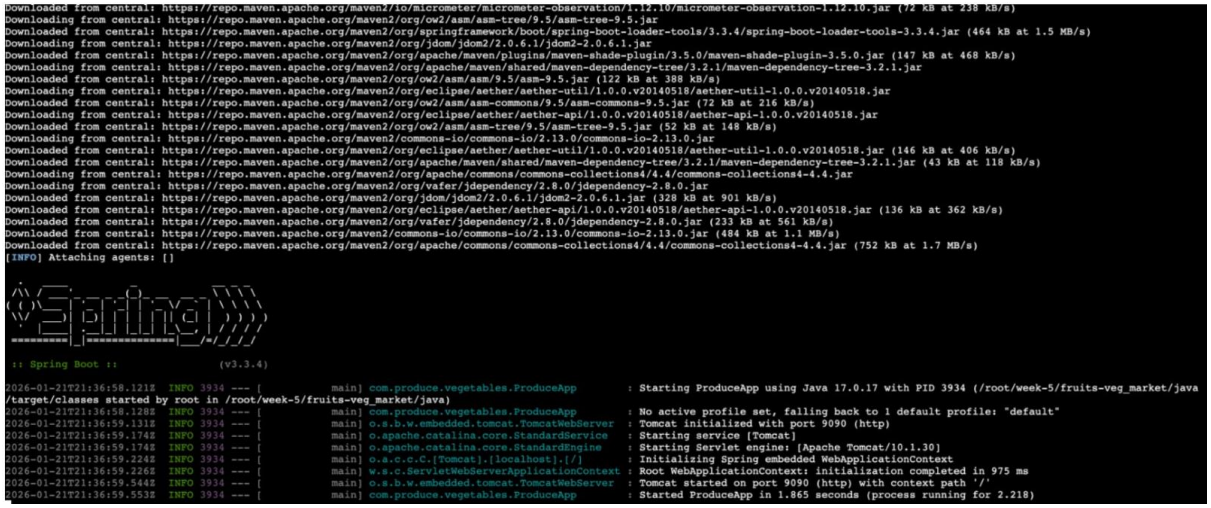
Fruits (FastAPI)

Fruit	Price ($)
Apple	1.20
Banana	0.80
Pineapple	2.20
Cherry	3.50
Orange	1.80
Blueberry	2.50
Lime	5.50
Mango	2.40
Kiwi	6.50
Organinc Cucumber	2.00
Lemon	2.00
Grapes	2.20

Vegetables (Java)

Vegetable	Price ($)
Carrot	0.60
Broccoli	1.40
Tomato	0.90
Spinach	1.10
Potato	0.50

Java running on port 9090



GitHub

https://github.com/odufuwabusola/Two-Tier-Infra-Packer