

Digital Egypt Pioneers Initiative

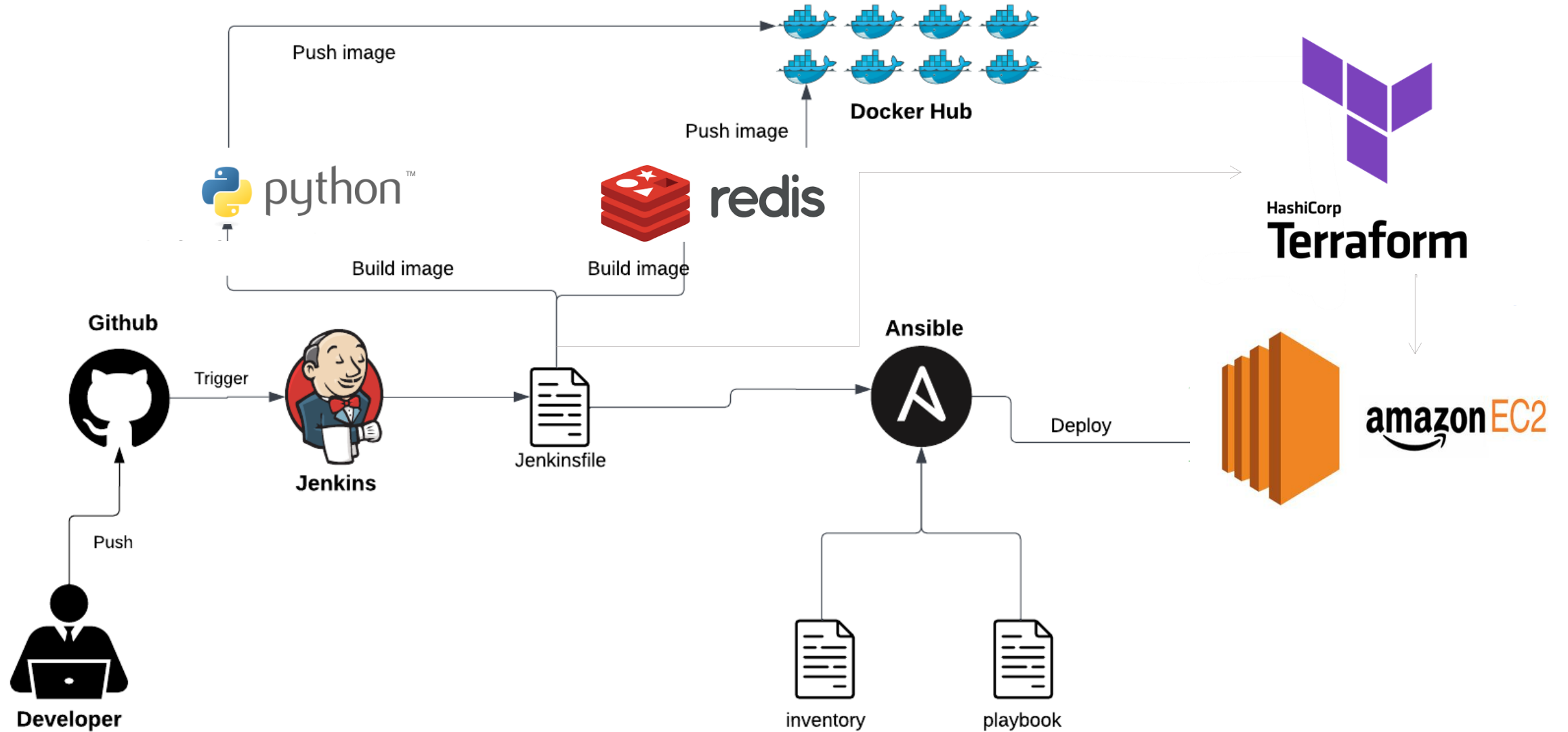
AUTOMATED DEPLOYMENT PIPELINE WITH JENKINS AND DOCKER

Supervisor: Eng. Ahmed Nasr

TEAM



Project overview



Phase 1: Initial Setup & Dockerization

- This phase focused on setting up the local environment, building the Dockerized application, and testing locally.



```
version: '3'
services:
  app:
    build: .
    container_name: my_python_app
    ports:
      - "5000:5000"
    environment:
      - ENVIRONMENT=DEV
      - HOST=0.0.0.0
      - PORT=5000
      - REDIS_HOST=redis
      - REDIS_PORT=6379
      - REDIS_DB=0
    depends_on:
      - redis
    command: sh -c "sleep 5 && python hello.py" # Wait for Redis to start

  redis:
    image: "redis:5.0"
    container_name: redis
    ports:
      - "6379:6379"
    volumes:
      - redis_data:/data
```



```
FROM python:3.8-slim

WORKDIR /app

COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 5000

CMD ["python", "hello.py"]
```

Phase 1: Initial Setup & Dockerization

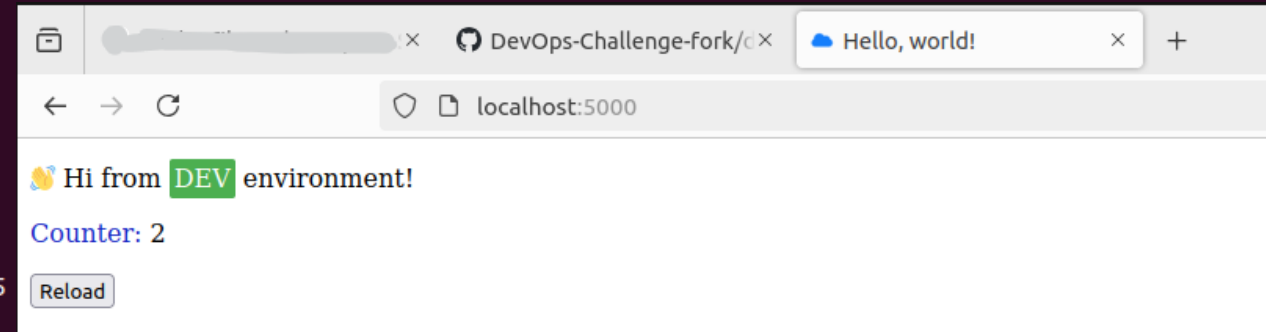
- This phase focused on setting up the local environment, building the Dockerized application, and testing locally.

```
hp@Ubuntu-Khaleel:~/devops-project/DevOps-Challenge-fork$ docker-compose up -d
Creating network "devops-challenge-fork_default" with the default driver
Creating volume "devops-challenge-fork_redis_data" with default driver
Pulling redis (redis:5.0)...
5.0: Pulling from library/redis
a603fa5e3b41: Pull complete
77631c3ef092: Pull complete
ed3847cf62b8: Pull complete
295236254fbb: Pull complete
1d25d6f70191: Pull complete
23acdbbf5eef: Pull complete
Digest: sha256:fc5ecd863862f89f04334b7cbb57e93c9790478ea8188a49f6e57b0967d38c75
Status: Downloaded newer image for redis:5.0
Creating redis ... done
Creating my_python_app ... done
```

```
hp@Ubuntu-Khaleel:~/devops-project/DevOps-Challenge-fork$ docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
d79d6db7c28f	devops-challenge-fork_app	my_python_app	"python hello.py"	9 seconds ago	Up 6 seconds	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
b4ded7136856	redis:5.0	redis	"docker-entrypoint.s..."	13 seconds ago	Up 10 seconds	0.0.0.0:6379->6379/tcp, :::6379->6379/tcp
d3b4cc686bc8	myjenkins-blueocean:2.414.2	jenkins-blueocean	"/usr/bin/tini -- /u..."	3 weeks ago	Up 16 minutes	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp

```
hp@Ubuntu-Khaleel:~/devops-project/DevOps-Challenge-fork$
```



Phase 2: Jenkins & CI Integration

- This phase we have to:
- Set up and configure Jenkins for CI/CD
- Integrate with Docker/GitHub/Dockerhub for automated builds

```
stages {  
  
    stage('Build and Push Docker Image') {  
        steps {  
            script {  
  
                sh 'docker build -t my_python_app .'  
                withCredentials([usernamePassword(credentialsId: 'dockerhub-credentials', passwordVariable: 'pass', usernameVariable: 'dockerhubuser')]) {  
                    sh 'docker login -u $dockerhubuser -p $pass'  
                    sh 'docker tag my_python_app:latest $DOCKER_IMAGE_NAME:latest'  
                    sh 'docker push $DOCKER_IMAGE_NAME:latest'  
                }  
            }  
        }  
    }  
  
    stage('Run App') {  
        steps {  
            script {  
                sh '''  
                # Clean up any existing containers  
                docker rm -f my_python_app || true  
                docker rm -f redis || true  
                # Run the application  
                docker-compose up -d  
                '''  
            }  
        }  
    }  
}
```

The screenshot shows the Jenkins web interface in a browser. The address bar indicates the URL is localhost:8080/job/final%20project/22/pipeline-graph/. The Jenkins logo and name are visible at the top. Below the navigation bar, the breadcrumb trail reads 'Dashboard > final project > #22 > Pipeline Overview'. The main heading is 'Build #22' with a green checkmark icon. The 'Pipeline' section displays a horizontal flowchart with steps: Start, Clone github re..., Build and Push ..., Run App, Test, Deploy, Post Actions, and End. Each step is marked with a green checkmark, indicating successful completion. The 'Run App' step is currently selected.

[▶ Rebuild](#) [📋 Console](#) [⚙️ Configure](#)

Details

👤 Manually run by Ola Youssef Mohammed
🕒 Started 55 sec ago
📅 Queued 5 ms
⌚ Took 37 sec

Phase 2: Jenkins & CI Integration

- Build Docker images and run automated tests, configure notifications for build results
- Push Docker images to Docker Hub or a private registry
- Automate application deployment.

Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

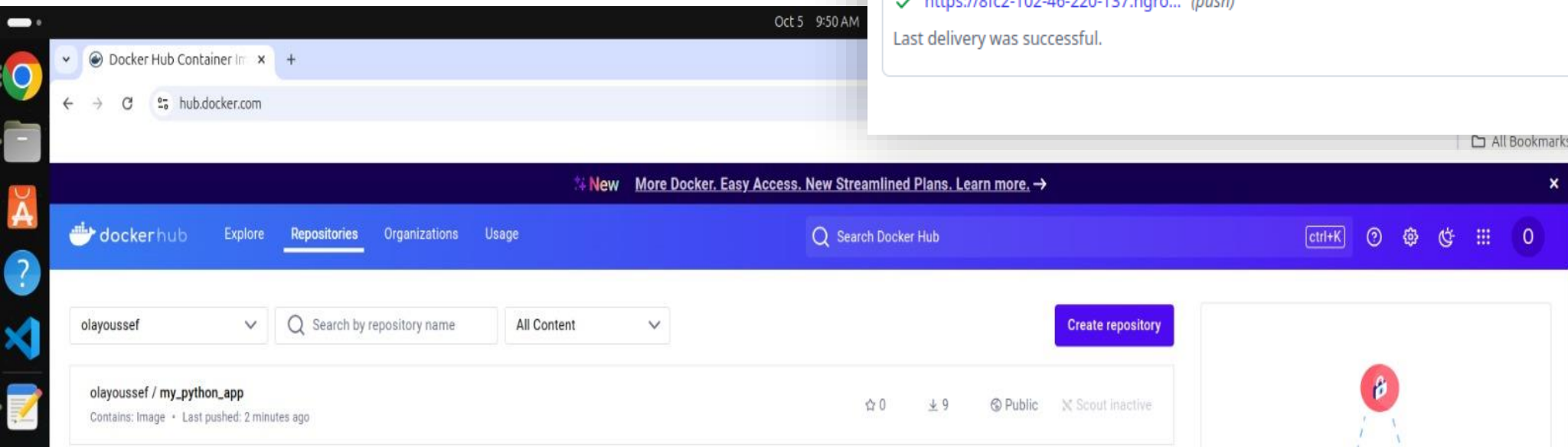
We will also send events from this repository to your [organization webhooks](#).

✓ <https://8fc2-102-46-220-137.ngro...> (push)


Edit

Delete

Last delivery was successful.



Phase 2: Jenkins & CI Integration

 **Jenkins**

Dashboard > final project > #22

Status

Changes

Console Output

Edit Build Information

Delete build '#22'

Timings

Git Build Data

Pipeline Overview

Pipeline Console

Restart from Stage

Replay

Pipeline Steps

Workspaces

Previous Build

✓ Console Output

Started by user [Ola Youssef Mohammed](#)
[Pipeline] Start of Pipeline
[Pipeline] node
Running on [Jenkins](#) in /var/lib/jenkins/workspace/final project
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone github repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/final project/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url <https://github.com/DEPI-DevOps-tasks/DevOps-Challenge-fork> # timeout=10
Fetching upstream changes from <https://github.com/DEPI-DevOps-tasks/DevOps-Challenge-fork>
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- <https://github.com/DEPI-DevOps-tasks/DevOps-Challenge-fork> +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision c84619fc6b8628b8b6de288997213ef46ca9195c (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f c84619fc6b8628b8b6de288997213ef46ca9195c # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D master # timeout=10
> git checkout -b master c84619fc6b8628b8b6de288997213ef46ca9195c # timeout=10
Commit message: "Updated docker-compose.yml"
> git rev-list --no-walk c84619fc6b8628b8b6de288997213ef46ca9195c # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build and Push Docker Image)

✓ Build #41 >

Success 13 min ago in 34 sec

✓ Checkout SCM

✓ Build and Push Docker Image

✓ Run App

✓ Test

✓ Deploy

✓ Post Actions

Phase 3: Infrastructure Provisioning with Terraform

- This phase used Terraform to provision AWS resources for the application's deployment.

The screenshot displays the Jenkins pipeline interface for a pipeline named 'new < 5'. The browser address bar shows 'localhost:8080/blue/organizations/jenkins/new/detail/new/5/pipeline'. The interface includes tabs for 'Pipeline', 'Changes', 'Tests', and 'Artifacts', along with a 'Logout' button. A green header bar contains the pipeline name and a 'Pipeline' tab. Below this, a summary bar shows 'Branch: -', 'Commit: -', a duration of '1m 59s', and the status 'No changes'. A green bar below the summary shows '3 minutes ago' and 'Started by user maram hassan'. The main area features a pipeline graph with steps: Start, Checkout Terrafor..., Install Terraform, Terraform Init, Terraform Plan, Terraform Apply, and End. All steps are marked with green checkmarks, and 'Terraform Apply' is highlighted with a blue circle. Below the graph, the 'Terraform Apply - 35s' step is expanded, showing a list of commands: 'terraform apply -auto-approve' (Shell Script, 33s) and 'Delete workspace when build is done' (<1s). A 'Restart Terraform Apply' button is visible in the top right of the expanded step.

new < 5

Pipeline Changes Tests Artifacts Logout

Branch: - 1m 59s No changes

Commit: - 3 minutes ago Started by user maram hassan

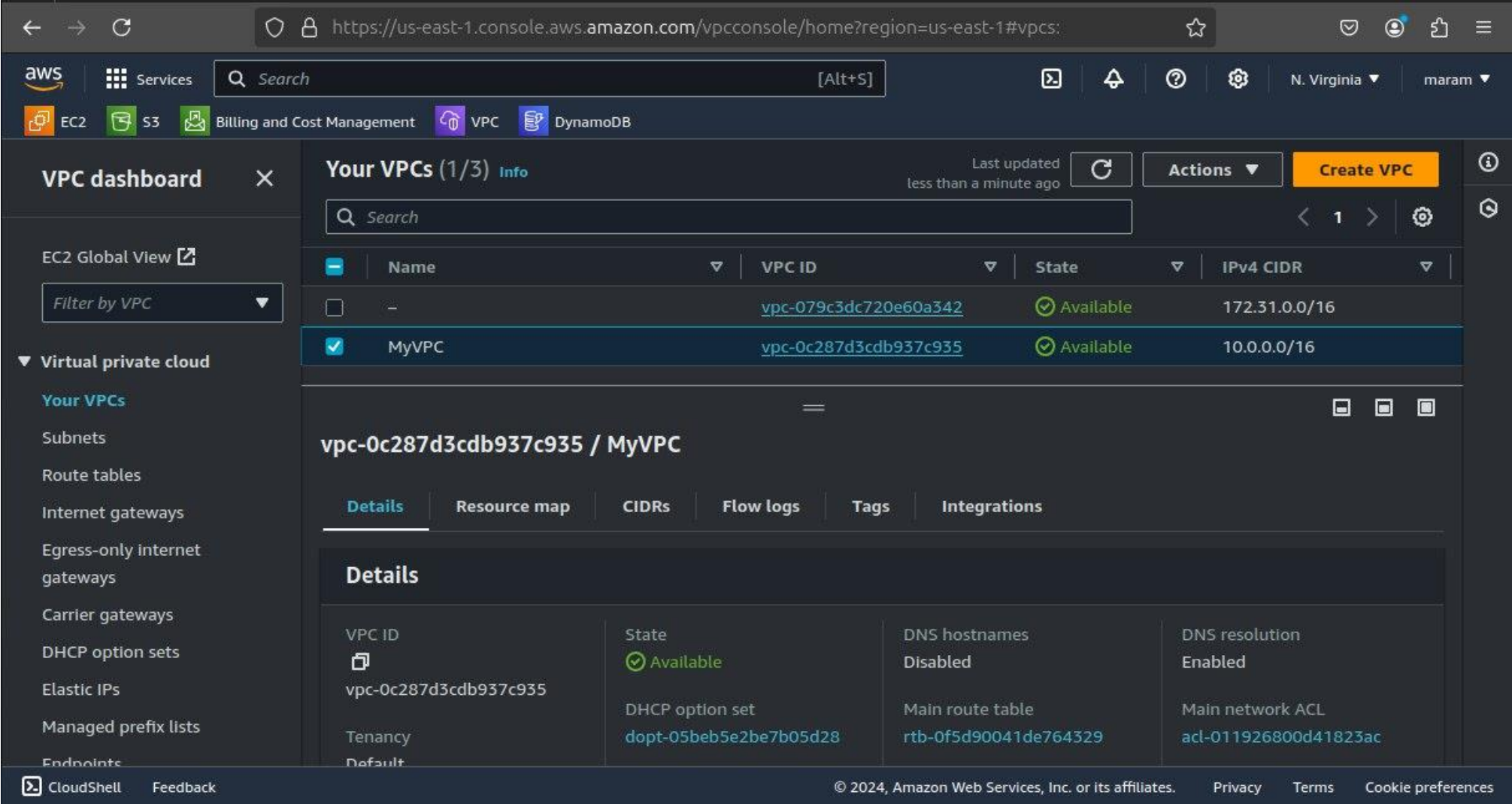
Start Checkout Terrafor... Install Terraform Terraform Init Terraform Plan Terraform Apply End

Terraform Apply - 35s

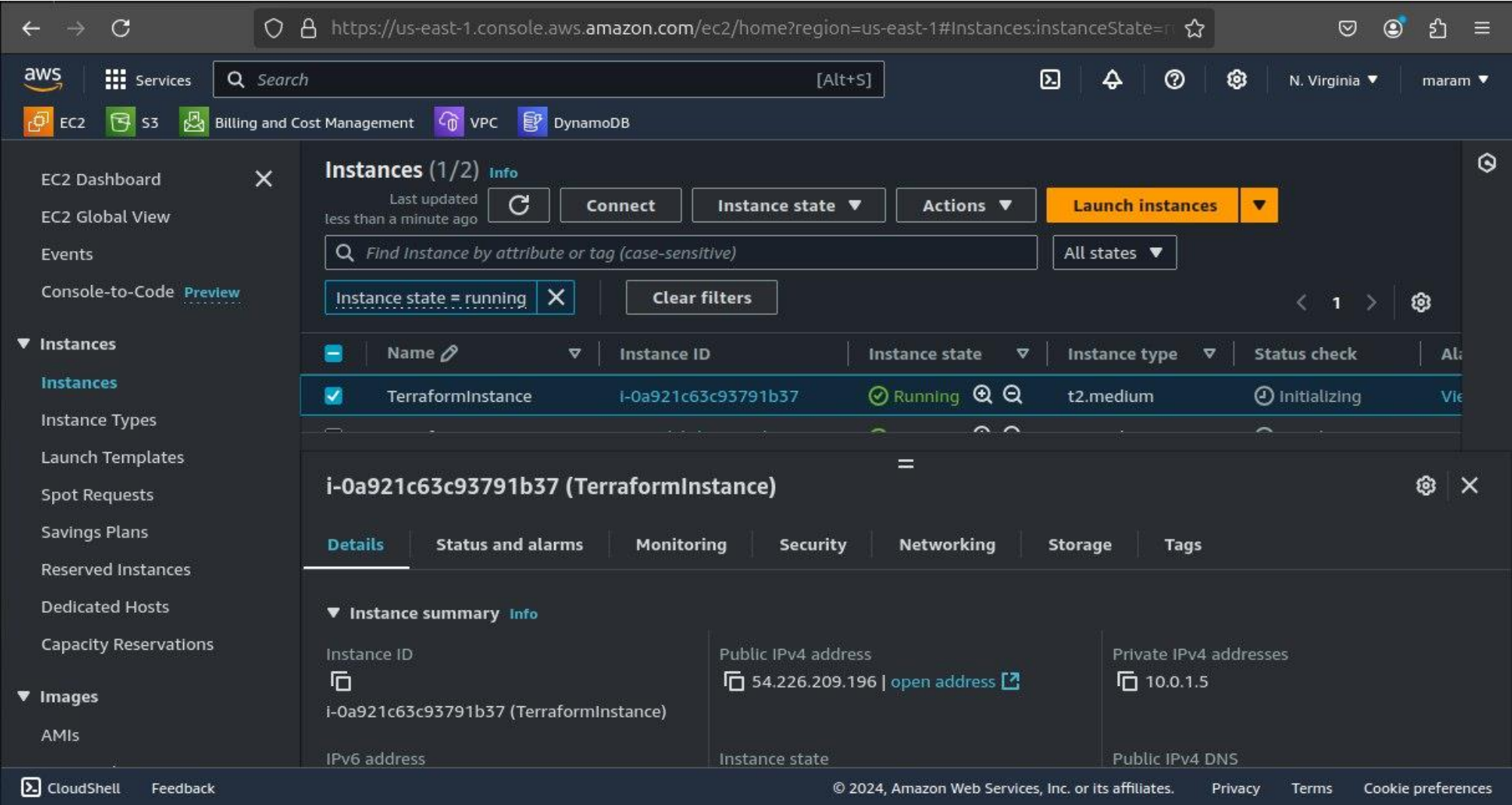
Restart Terraform Apply

- > terraform apply -auto-approve — Shell Script 33s
- > Delete workspace when build is done <1s

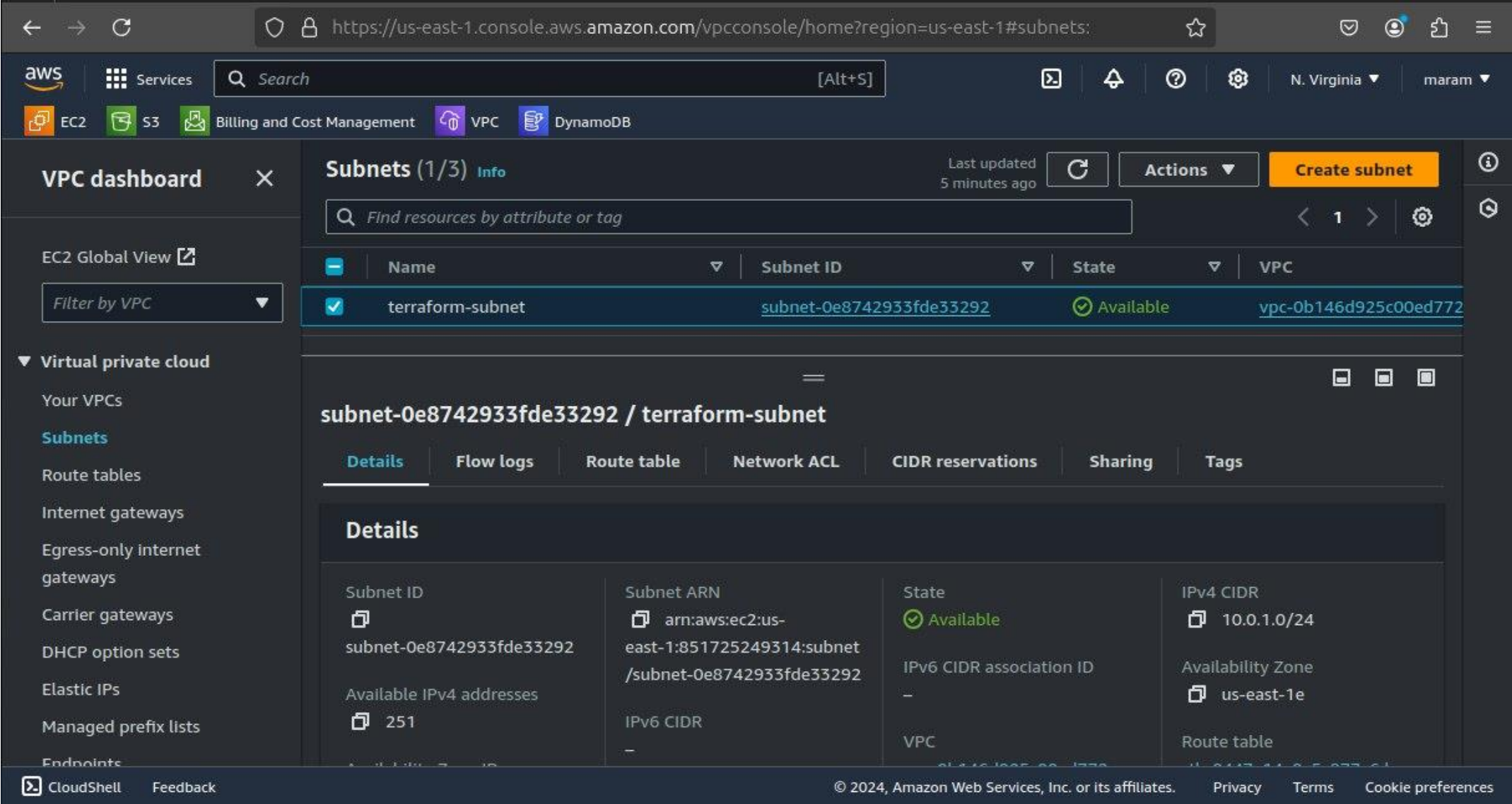
Phase 3: Infrastructure Provisioning with Terraform



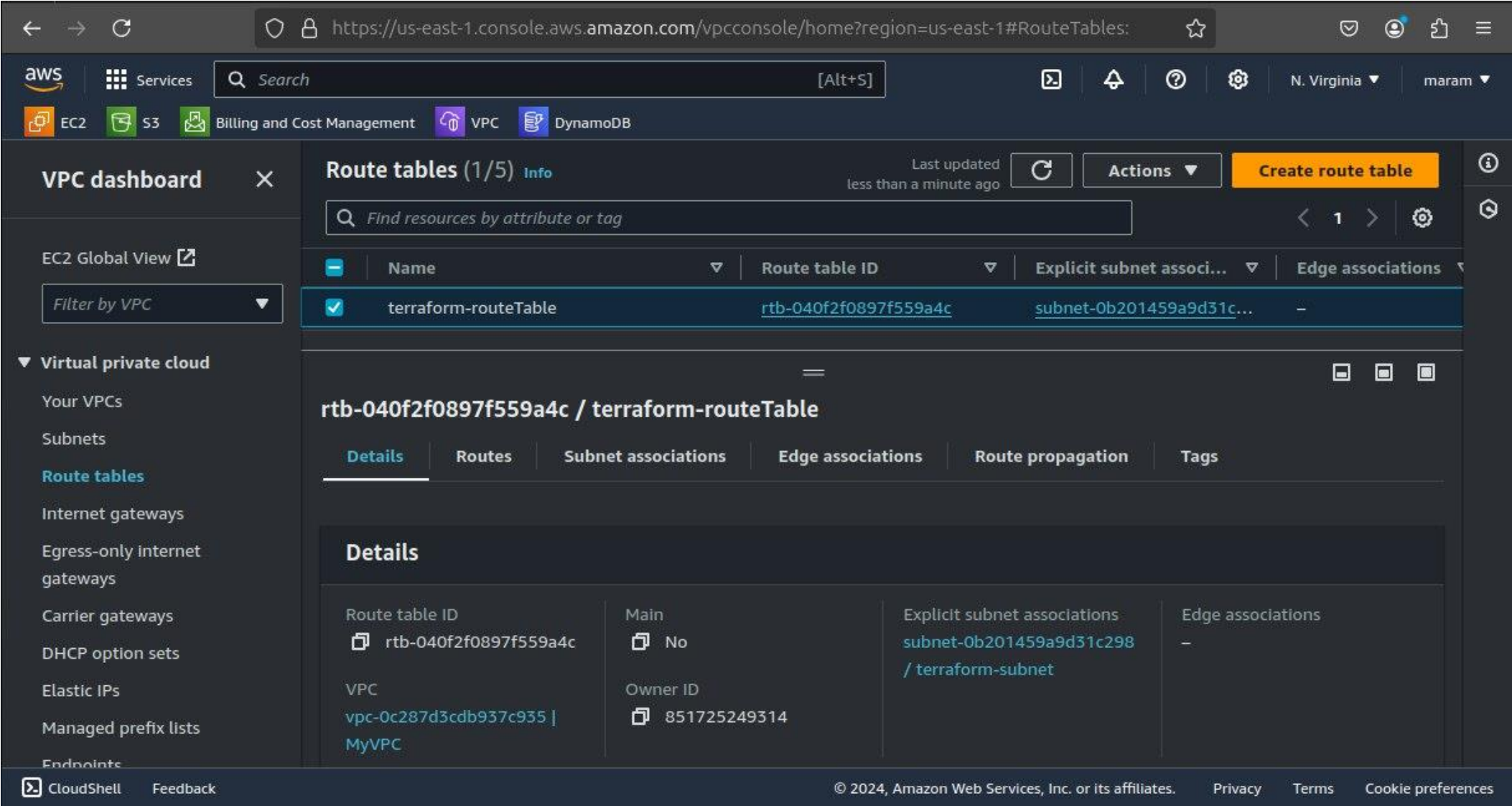
Phase 3: Infrastructure Provisioning with Terraform



Phase 3: Infrastructure Provisioning with Terraform



Phase 3: Infrastructure Provisioning with Terraform



Phase 4: Deployment with Ansible



```
- name: Deploy Docker Application on AWS EC2
hosts: all
become: true

tasks:
  - name: Gather facts
    setup:

  - name: Update apt and install required packages
    apt:
      name: "{{ item }}"
      state: present
    with_items:
      - docker.io
      - python3-pip
      - python3-venv
      - python3-apt
      - curl  # Ensure curl is installed
      - git  # Optional: Install git if you need versi...
```



```
- name: Start Docker service
  service:
    name: docker
    state: started
    enabled: true

- name: Create Python virtual environment
  command: python3 -m venv /home/ubuntu/venv
  args:
    creates: /home/ubuntu/venv

- name: Install Docker Python module in virtual environment
  command: /home/ubuntu/venv/bin/python -m pip install docker

- name: Create a directory for the application
  file:
    path: /home/ubuntu/app
    state: directory
```

Phase 4: Deployment with Ansible



```
- name: Copy docker-compose.yml to EC2 instance
  copy:
    src: ./docker-compose.yml
    dest: /home/ubuntu/app/docker-compose.yml

- name: Install Docker Compose
  shell: >
    curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)"
-o /usr/local/bin/docker-compose
  args:
    creates: /usr/local/bin/docker-compose

- name: Set permissions for Docker Compose
  command: chmod +x /usr/local/bin/docker-compose

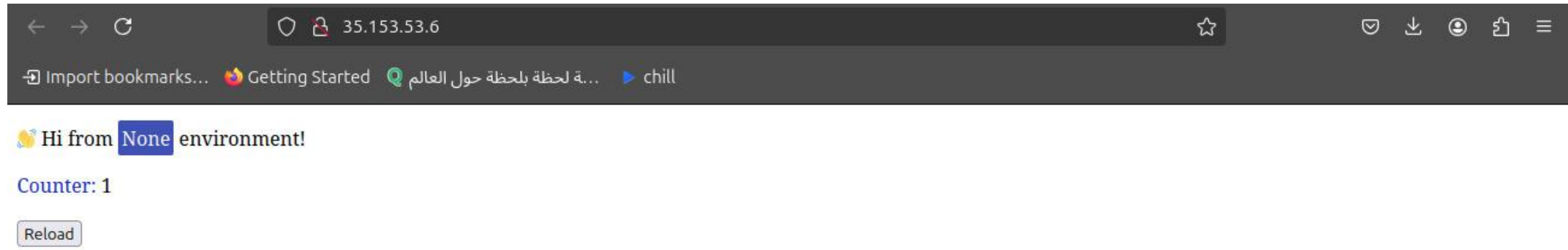
- name: Verify Docker Compose installation
  command: docker-compose --version
  register: docker_compose_version

- debug:
  var: docker_compose_version.stdout

- name: Pull Docker images
  command: docker-compose -f /home/ubuntu/app/docker-compose.yml pull
  args:
    chdir: /home/ubuntu/app

- name: Run Docker containers
  command: docker-compose -f /home/ubuntu/app/docker-compose.yml up -d
  args:
    chdir: /home/ubuntu
```

Phase 4: Deployment with Ansible



Digital Egypt Pioneers Initiative

THANK YOU

TEAM

