**Fatima Jinnah Women University**

***Department of Software Engineering***

|  |
| --- |
|  |
|  |  |

**SUBJECT: CC LAB**

**SUBMITTED TO: SIR SHOAIB**

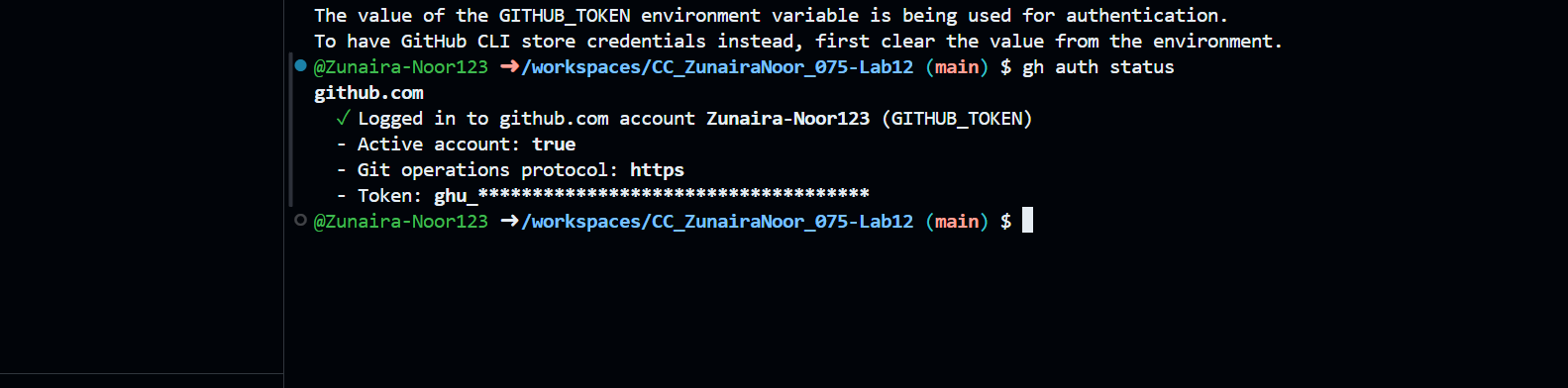
**SUBMITTED BY: ZUNAIRA NOOR**

**REGISTRATION NO: 2023-BSE-075**

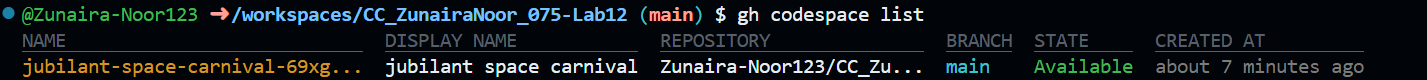
**SEMESTER: V-B**

**Task 0 Lab Setup (Codespace & GH CLI)**

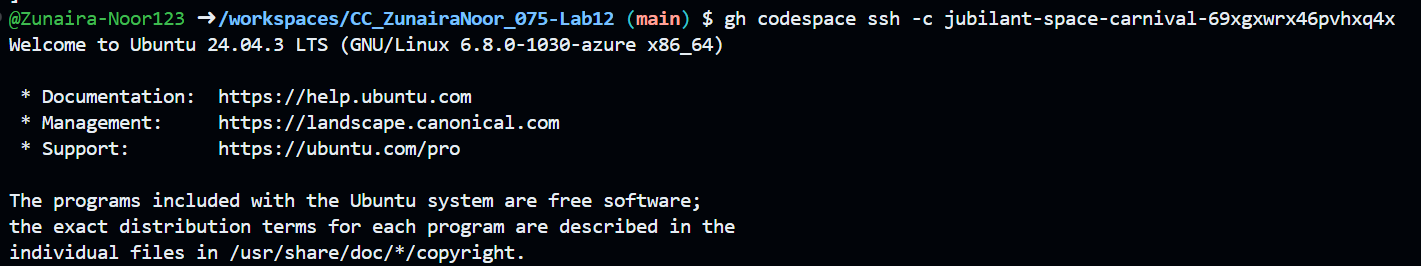
**Create Codespace & connect:**

****

**List Codespaces**

****

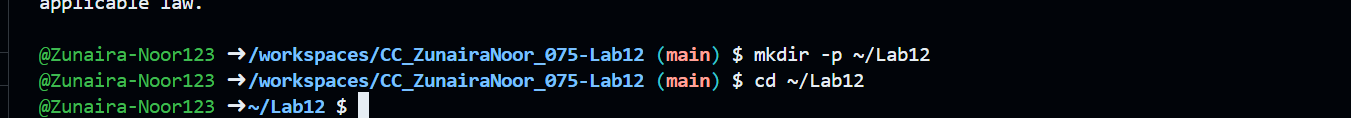
**SSH into Codespace**

****

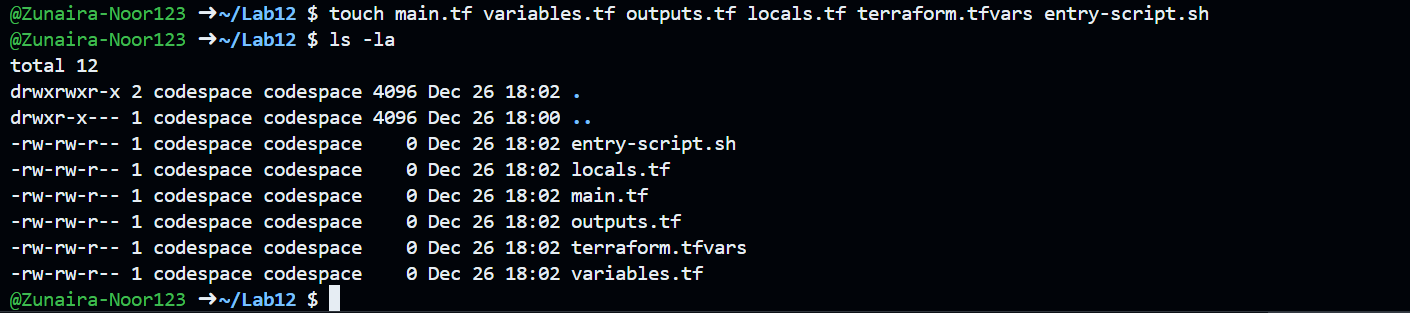
**Task 1 — Organize Terraform Code**

**Run ALL commands inside Codespace terminal**

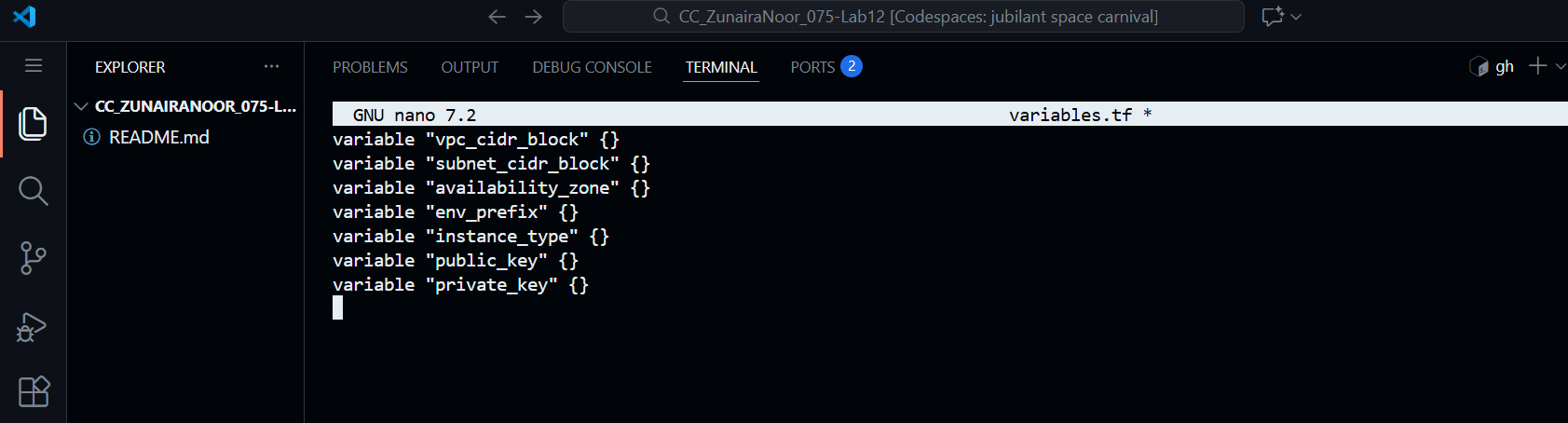
**Step 1: Create Project Directory**

****

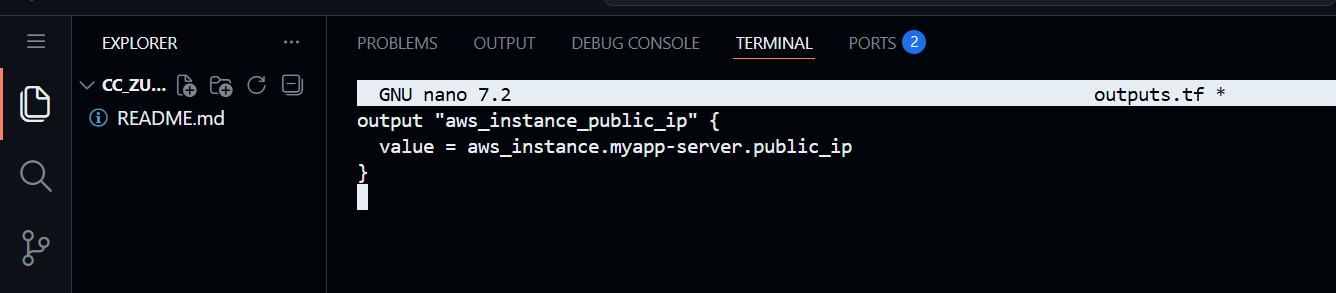
**Step 2: Create Required Files**

****

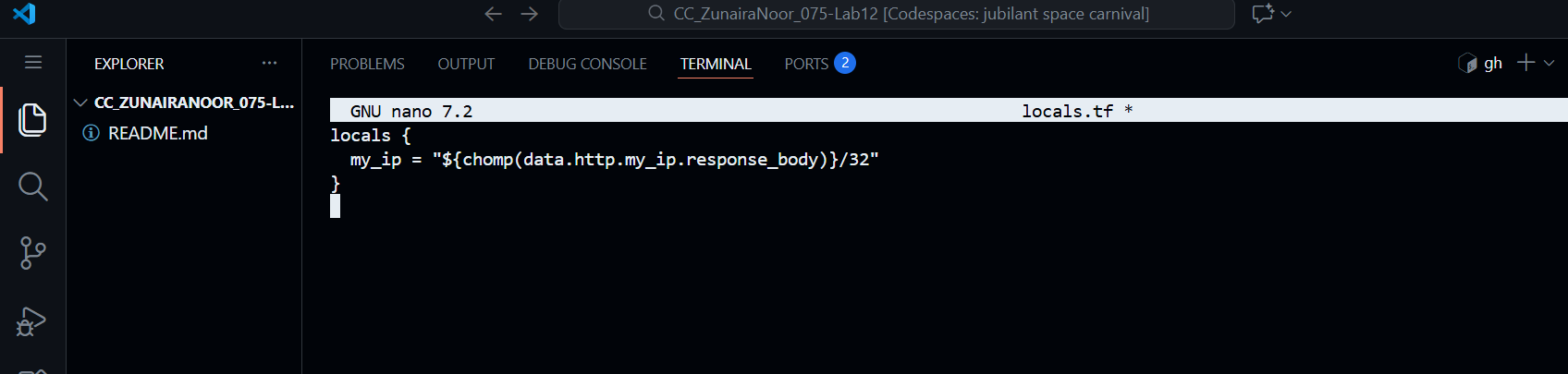
**Step 3: Create variables.tf**

****

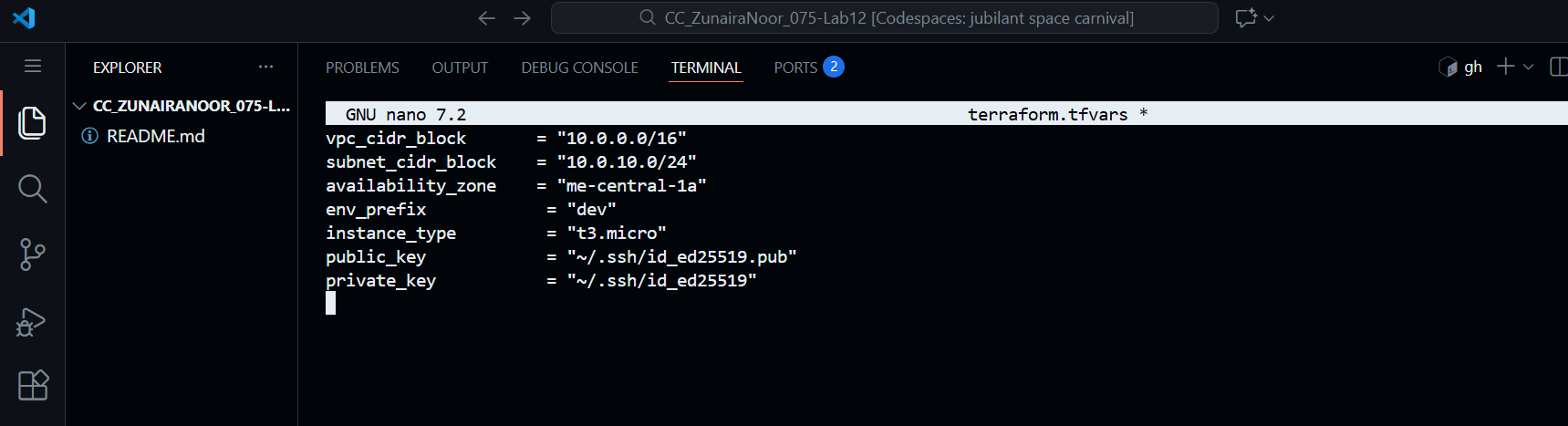
**Step 4: Create outputs.tf**

****

**Step 5: Create locals.tf**

****

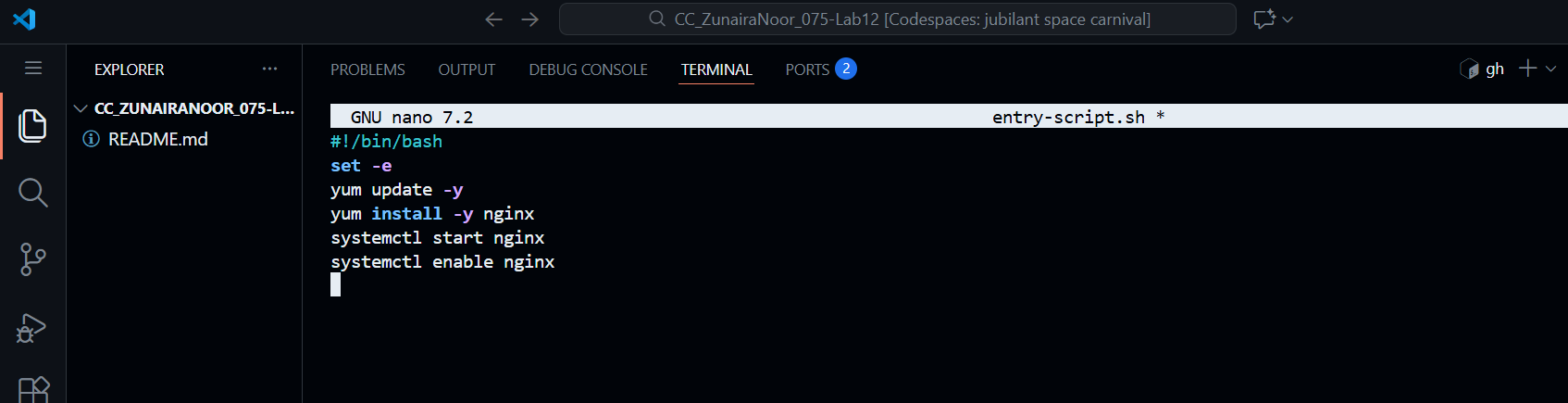
**Step 6: Create terraform.tfvars**

****

**Step 7: Create main.tf**

****

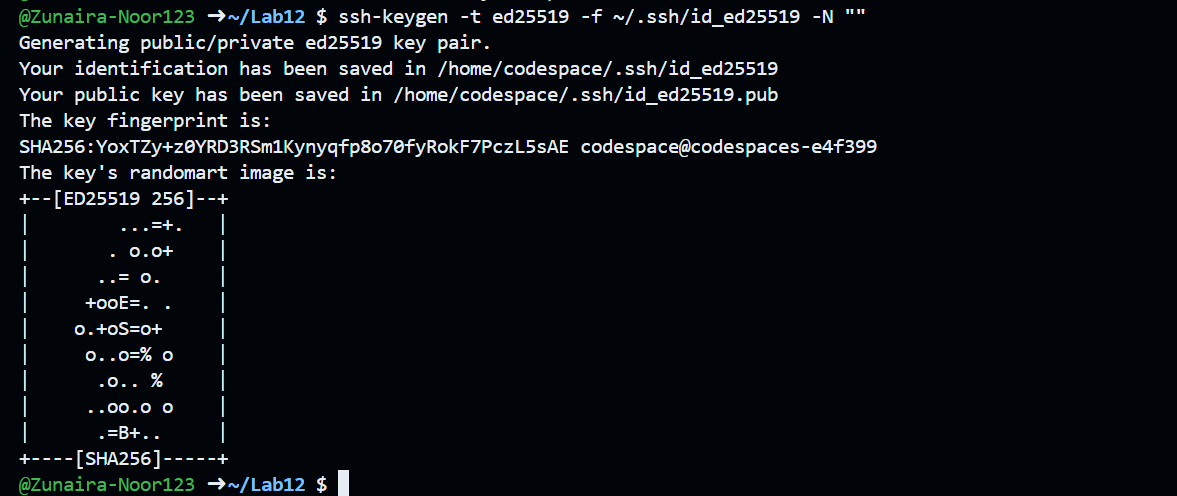
**Step 8: Create entry-script.sh**

****

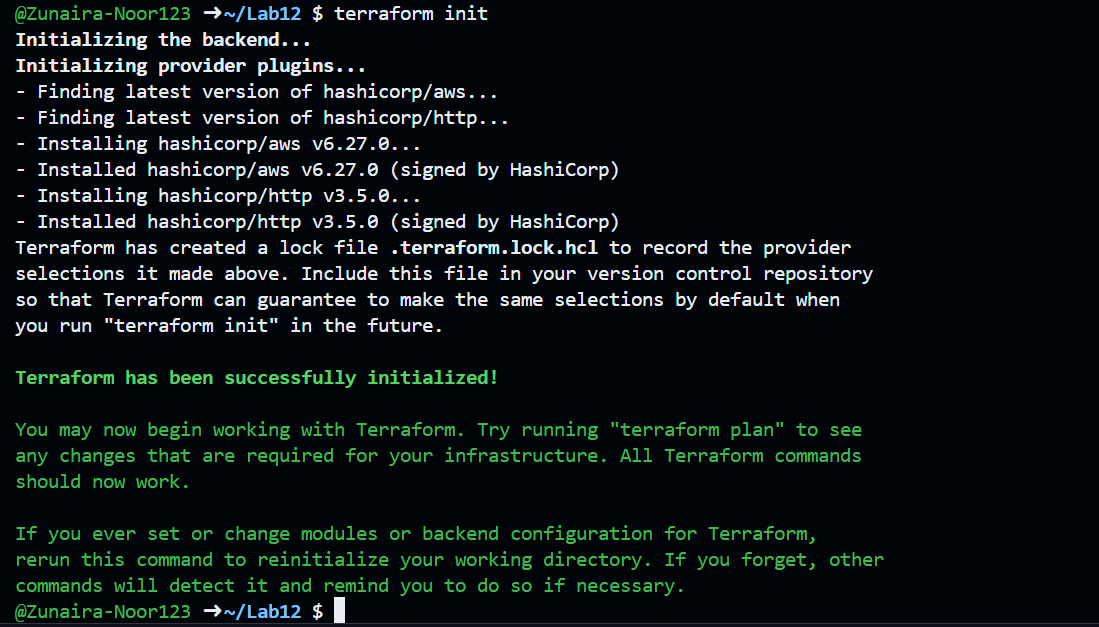
**Make executable:**

****

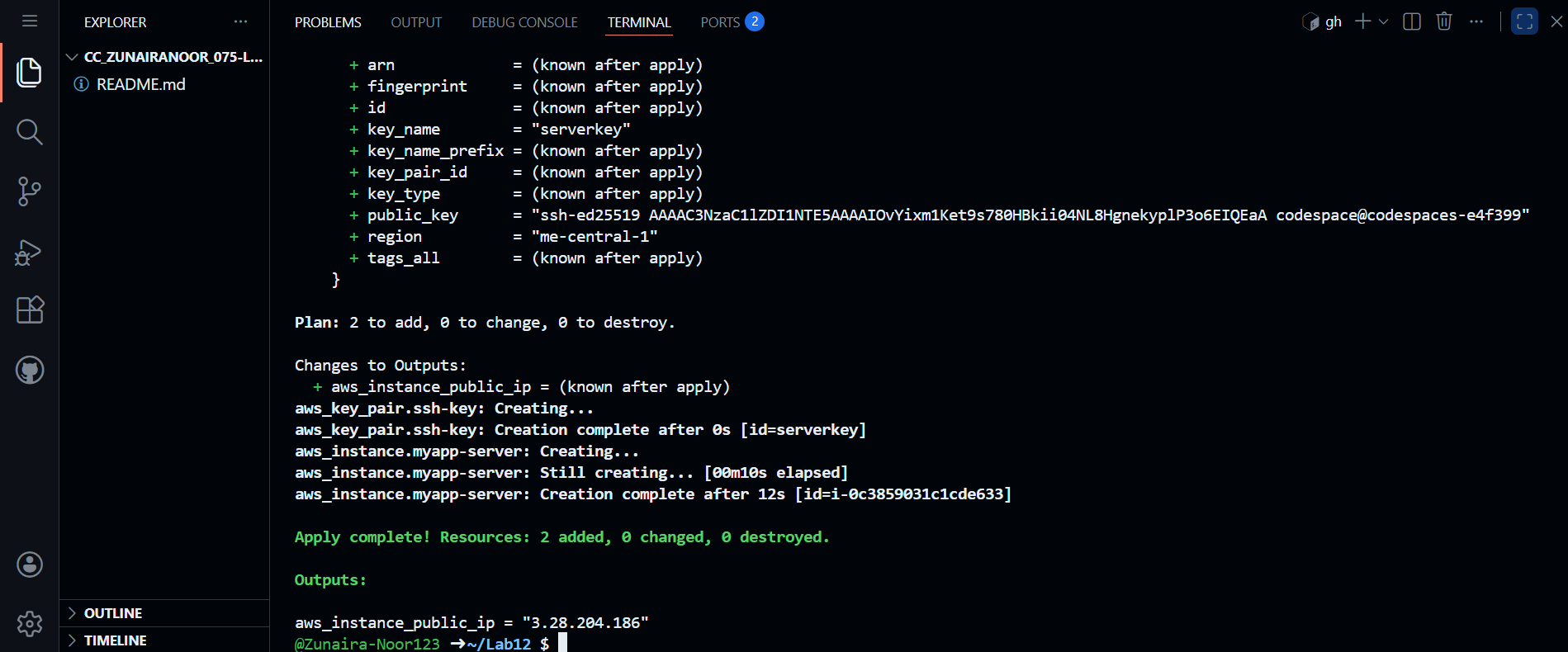
**Step 9: Generate SSH Key**

****

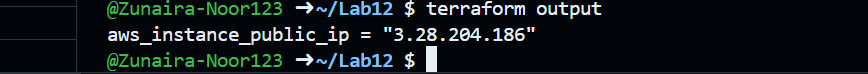
**Step 10: Initialize Terraform**

****

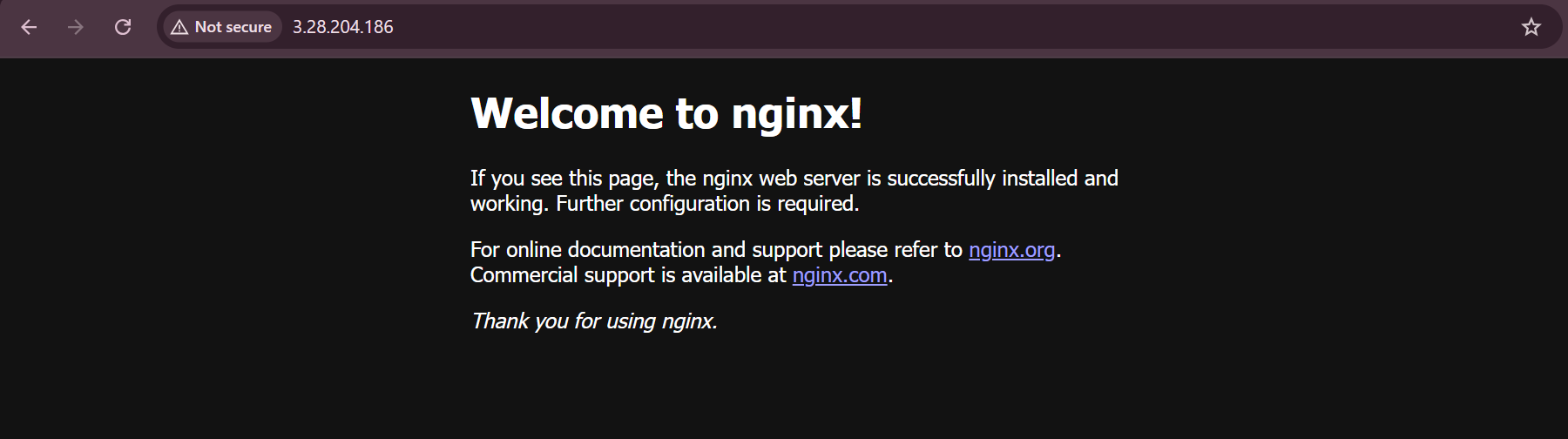
**Step 11: Apply Configuration**

****

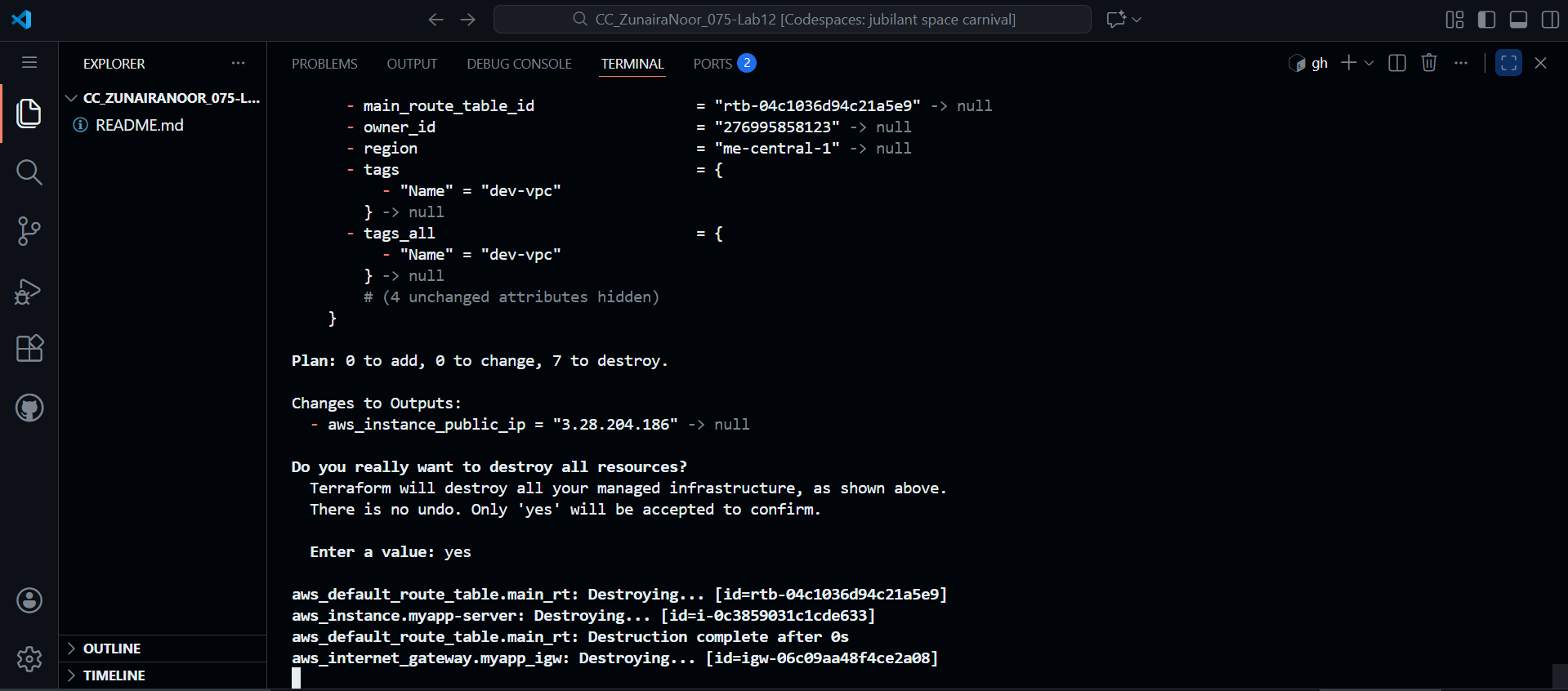
**After Apply Succeeds**

****

**Open browser:**

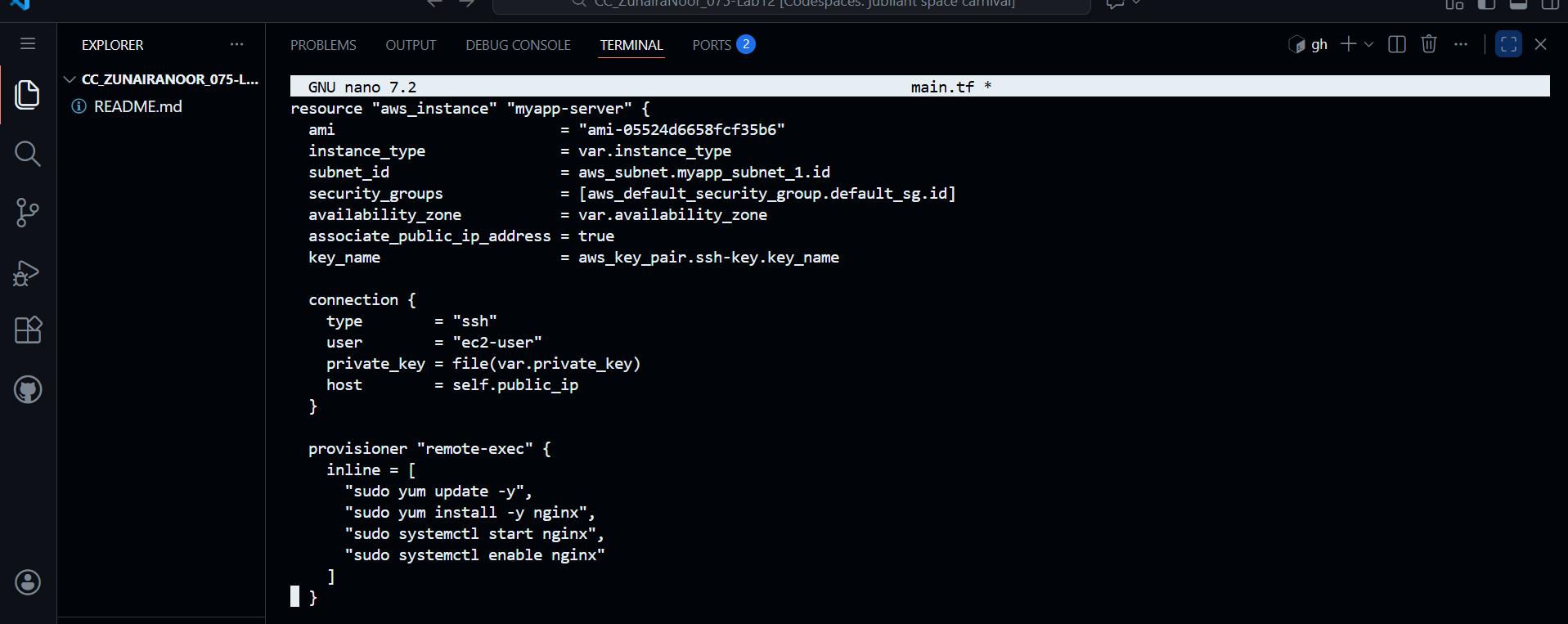
****

**Destroy Resources**

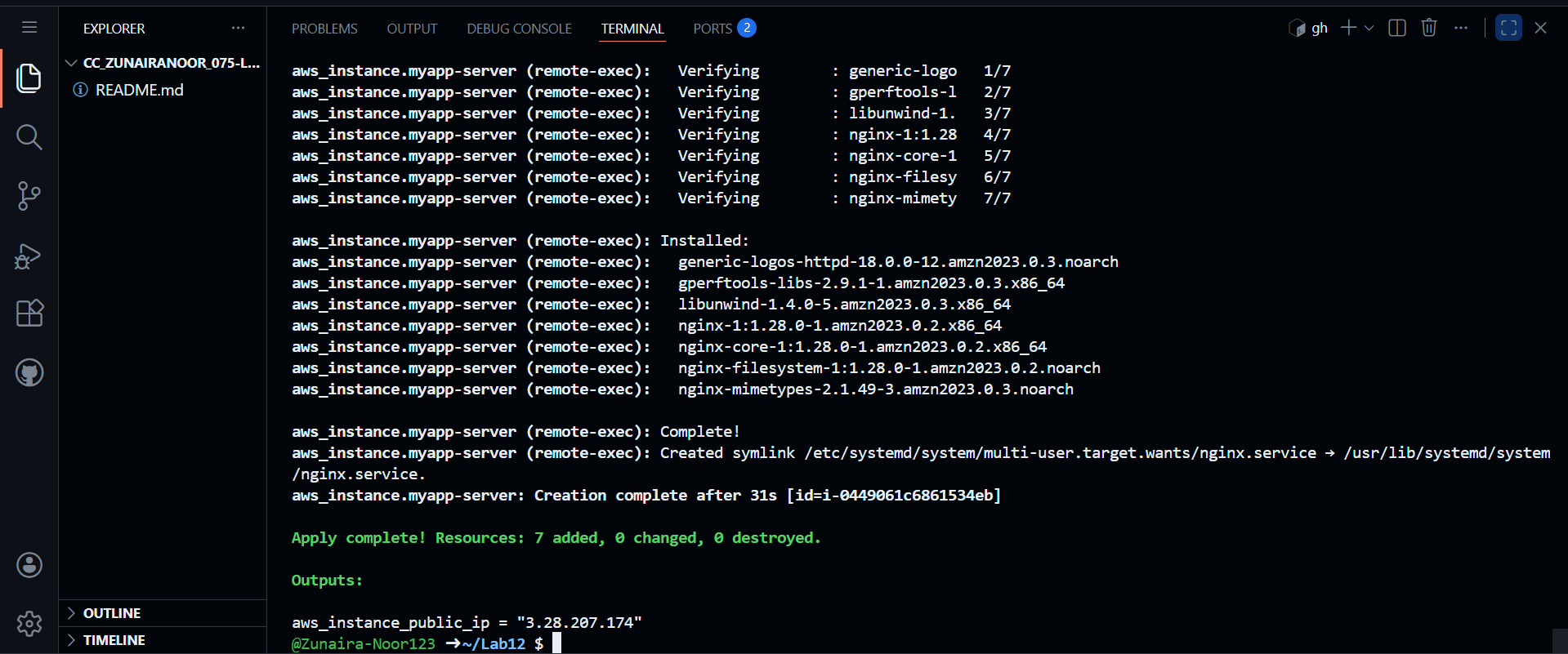
****

**Task 2 — Use remote-exec Provisioner**

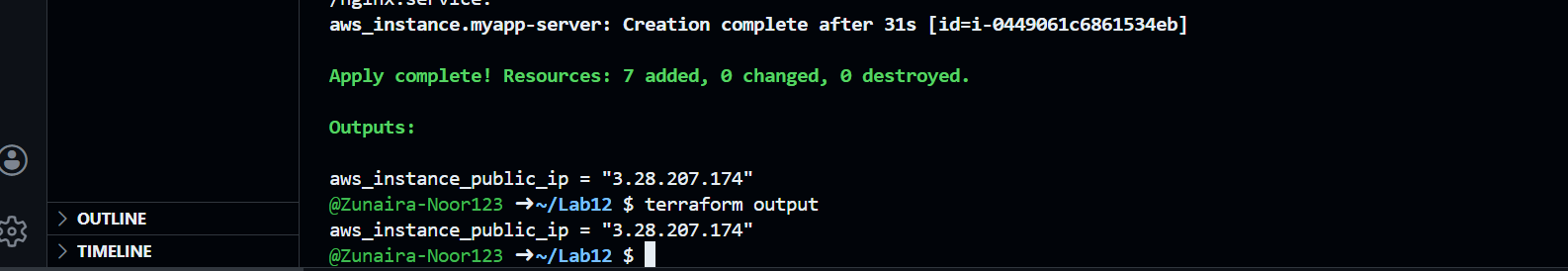
**Step 1: Edit main.tf**

****

**Step 2: Apply Terraform**

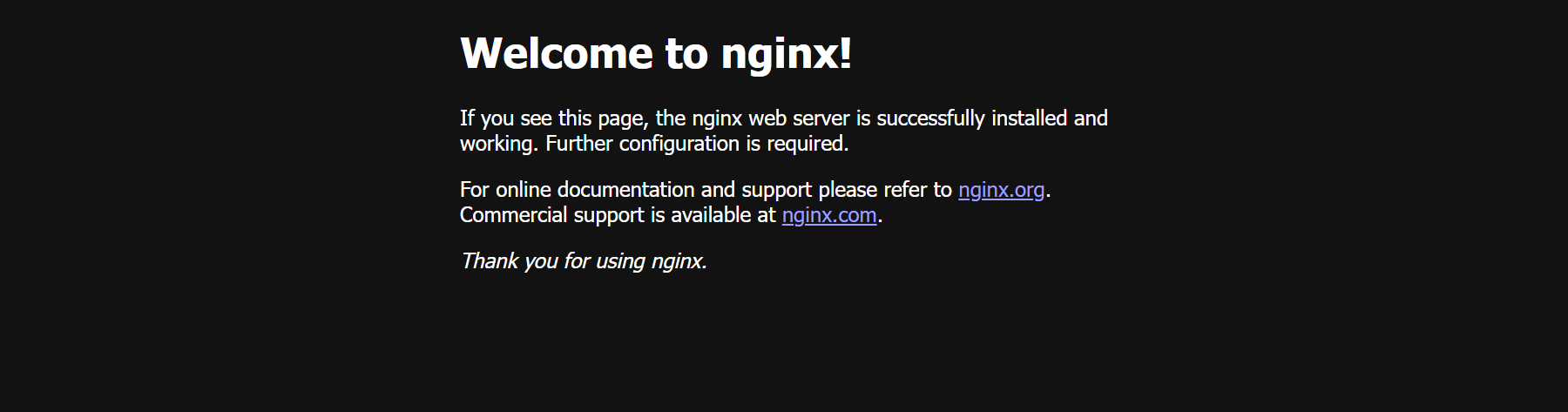
****

**Step 3: Display Output**

****

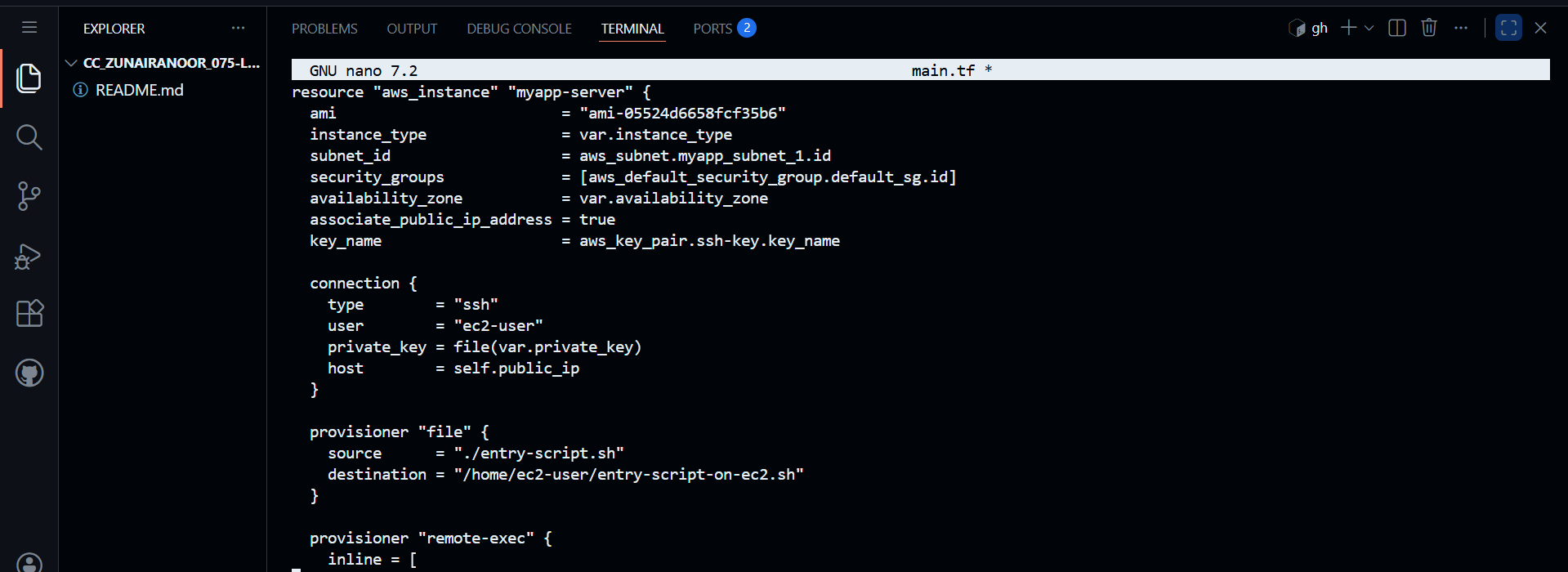
**Step 4: Test Nginx in Browser**

**Open browser and visit:**

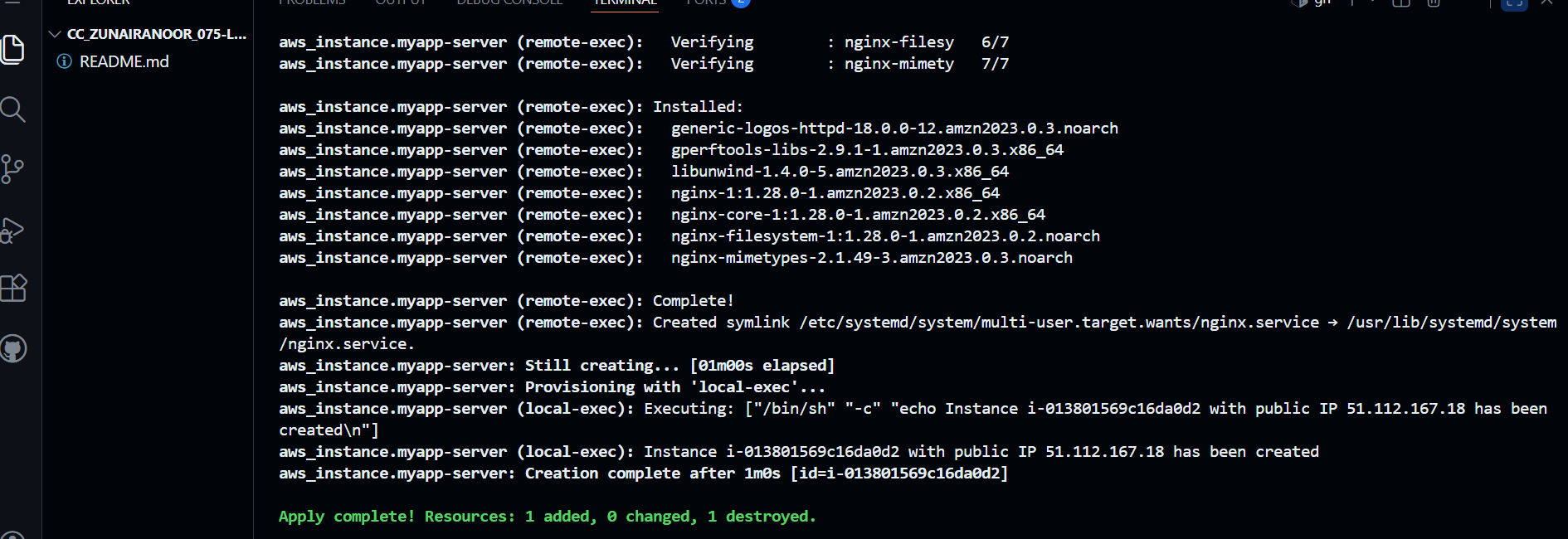
****

**Task 3 — File, Remote-exec & Local-exec Provisioners**

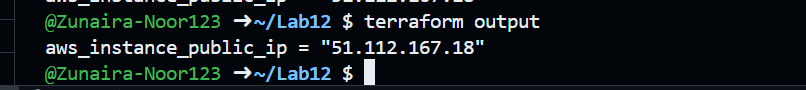
**Replace aws\_instance Resource**

****

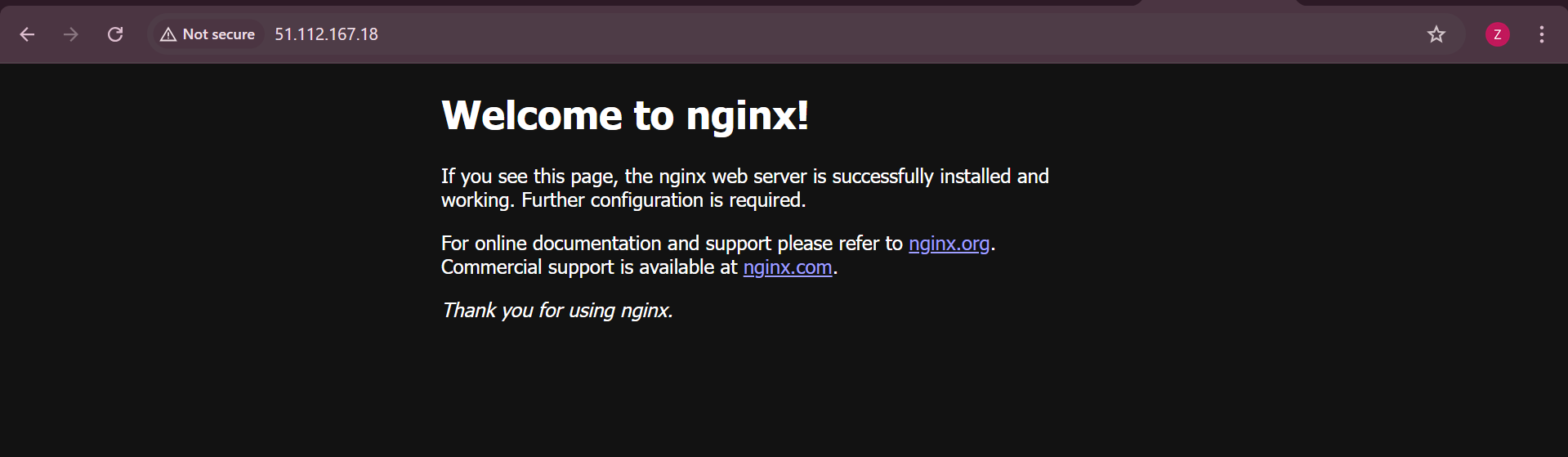
**Apply Terraform**

****

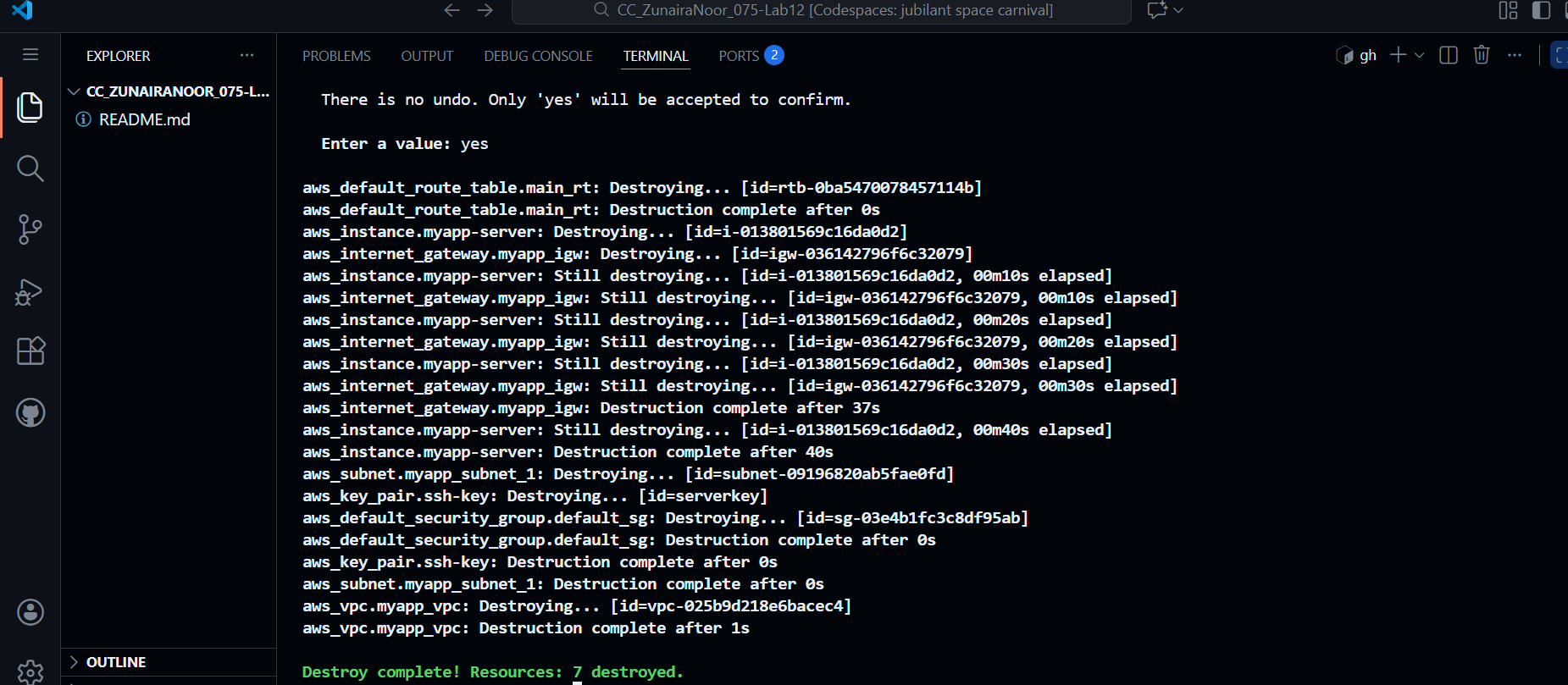
**Show Terraform Output**

****

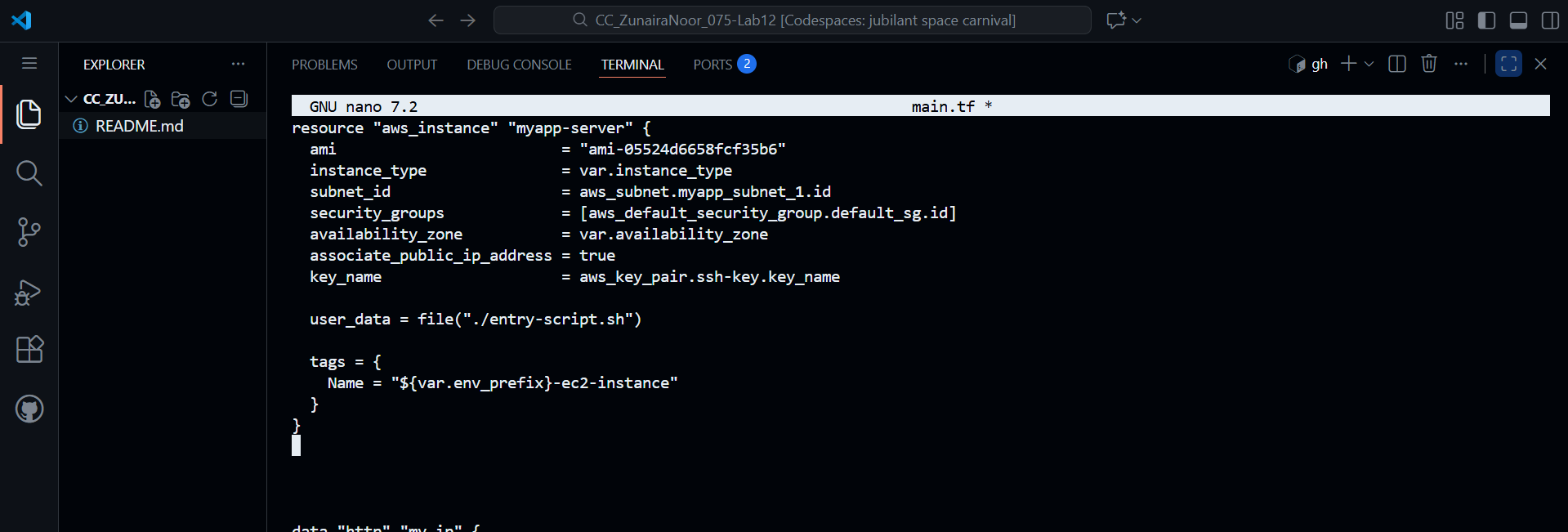
**Test Nginx in Browser**

****

**Destroy Resources**

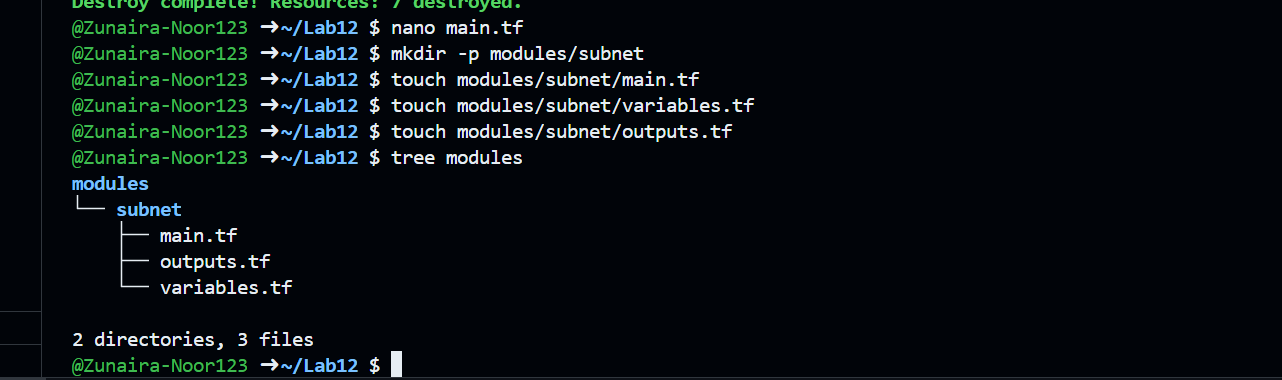
****

**Restore user\_data (Remove Provisioners)**

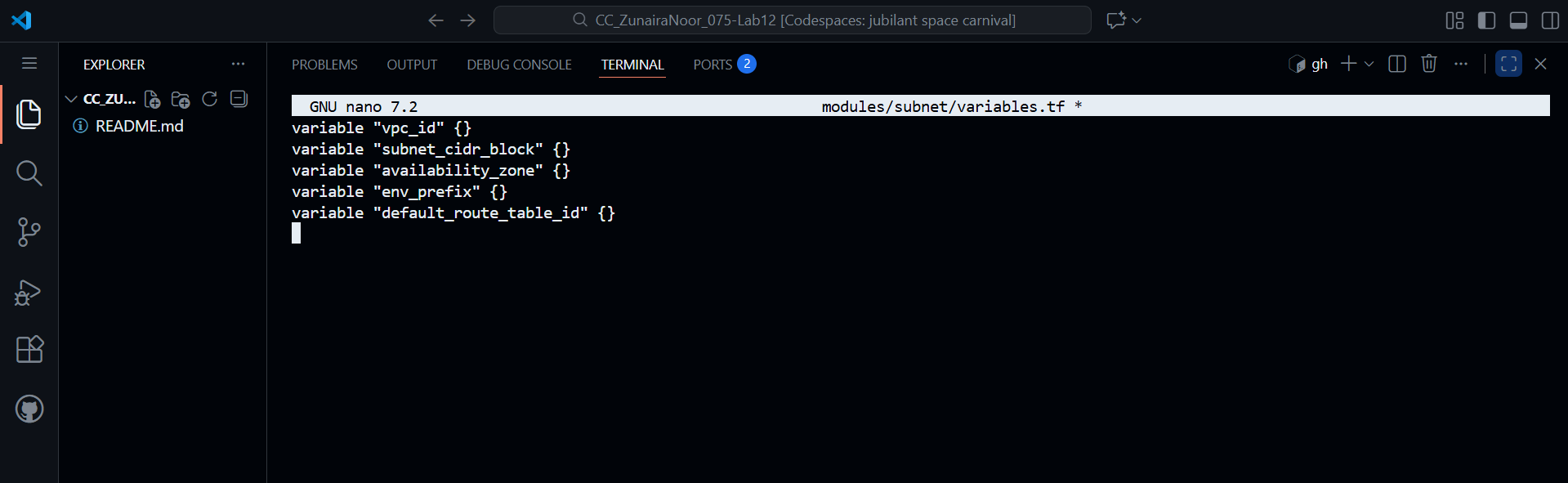
****

**Task 4 — Create Terraform Subnet Module**

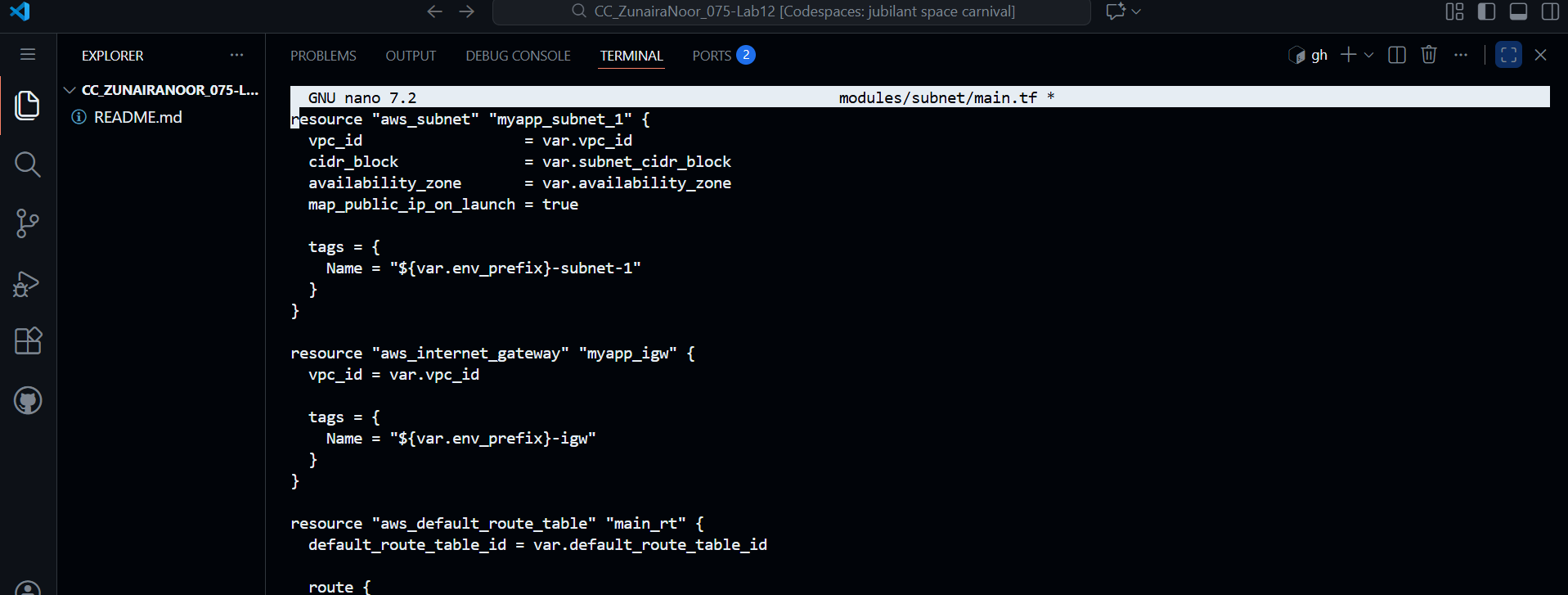
**Step 1: Create Module Directory Structure**

****

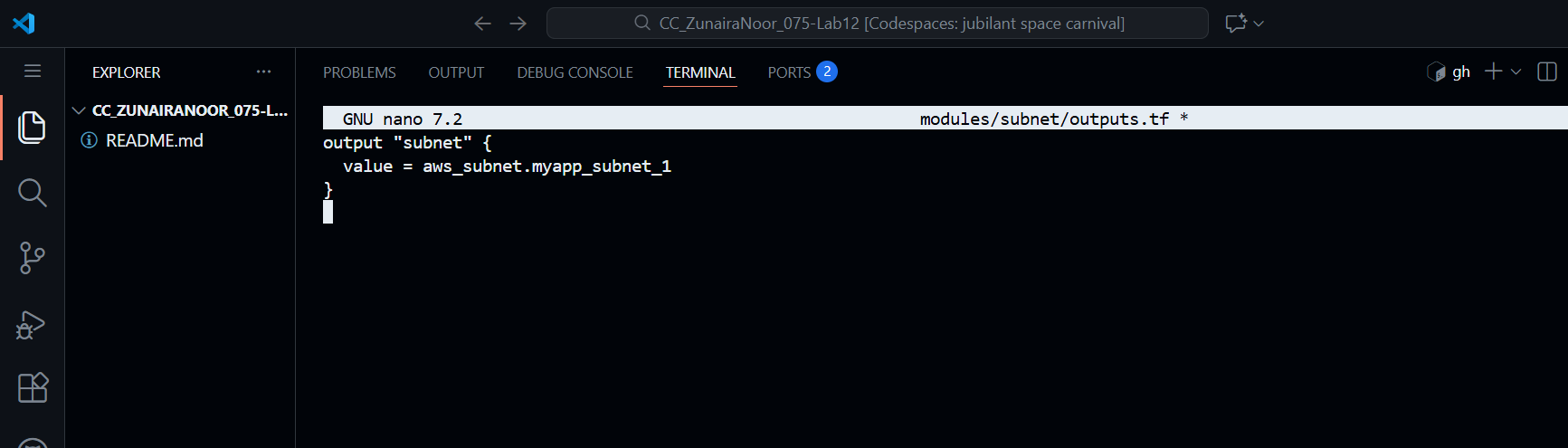
**Step 2: Create modules/subnet/variables.tf**

****

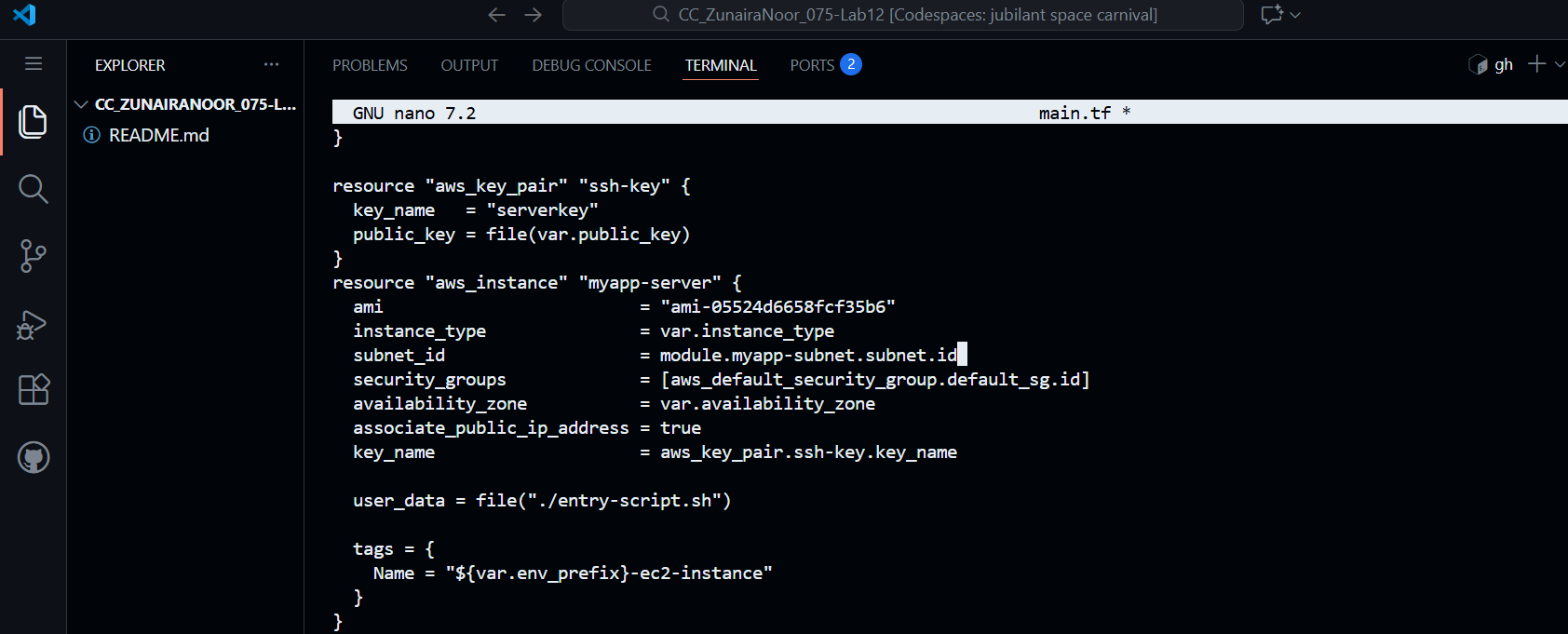
**Step 3: Create modules/subnet/main.tf**

****

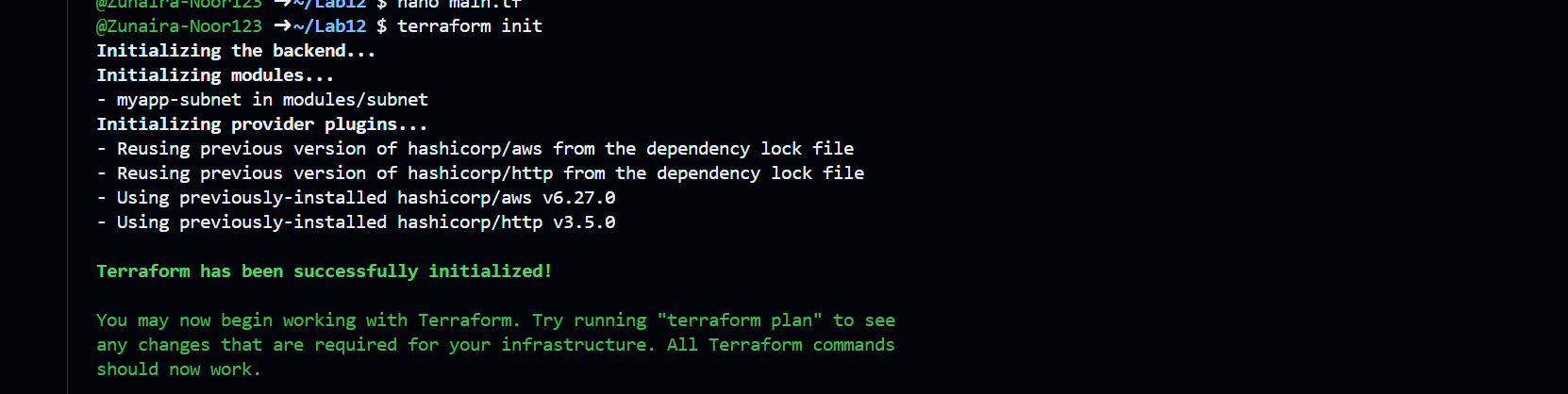
**Step 4: Create modules/subnet/outputs.tf**

****

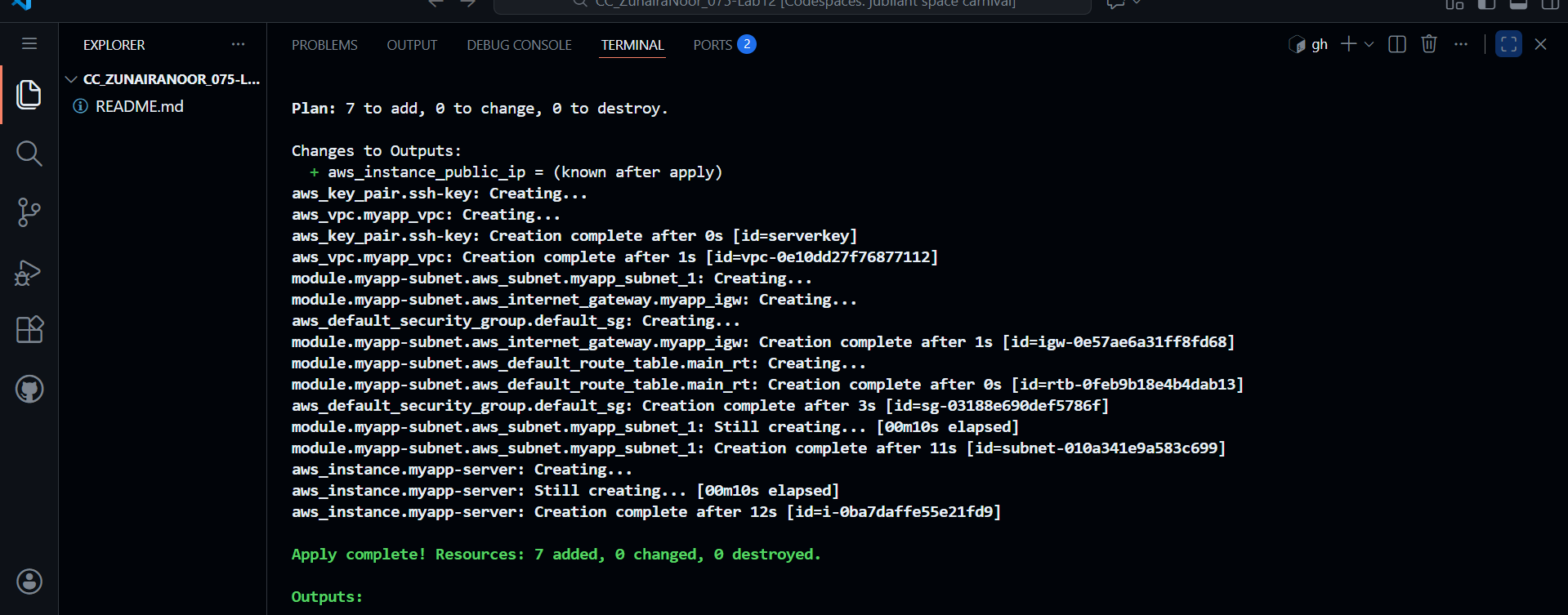
**Step 5: Modify Root main.tf to Use Module**

****

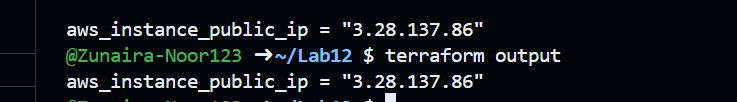
**Step 6: Initialize Terraform**

****

**Step 7: Apply Configuration**

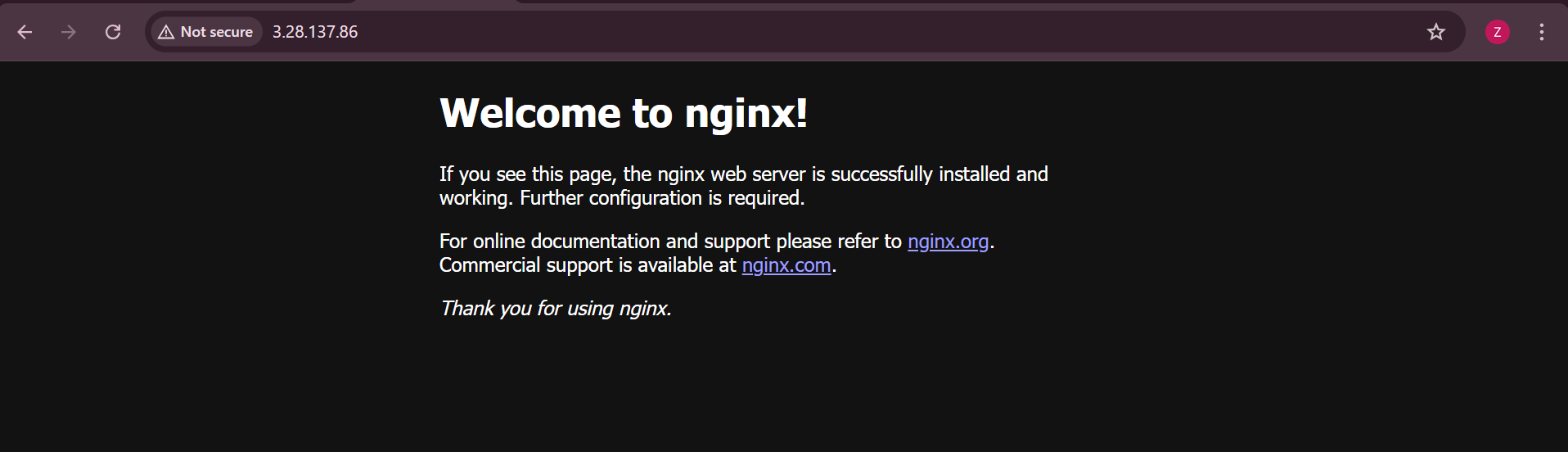
****

**Step 8: Show Output**

****

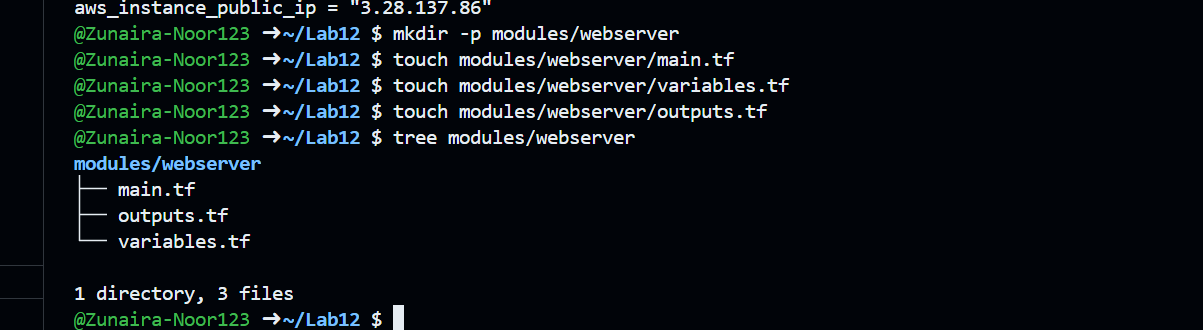
**Step 9: Test Nginx**

**Open browser:**

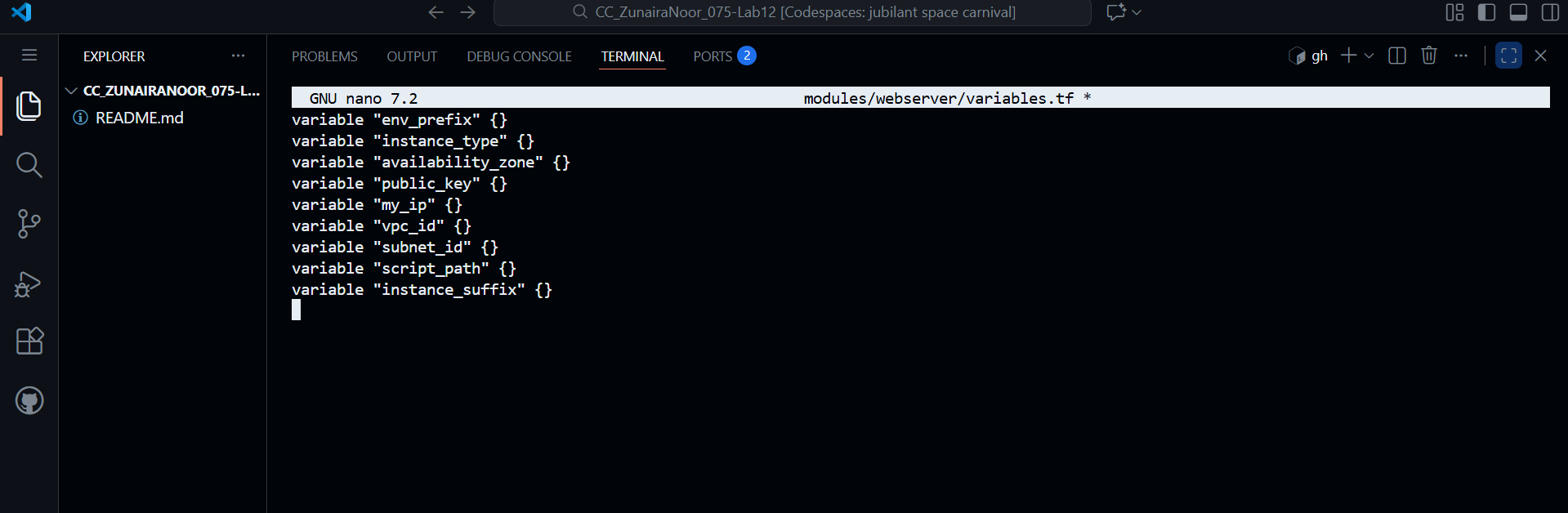
****

**Task 5 — Create Webserver Module**

**Step 1: Create Webserver Module Structure**

****

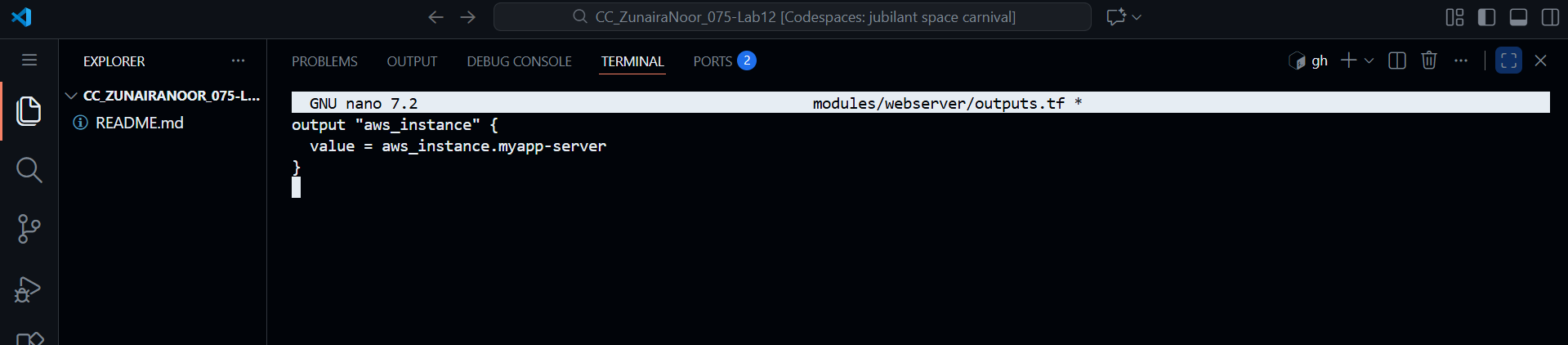
**Step 2: Create modules/webserver/variables.tf**

****

**Step 3: Create modules/webserver/main.tf**

****

**Step 4: Create modules/webserver/outputs.tf**

****

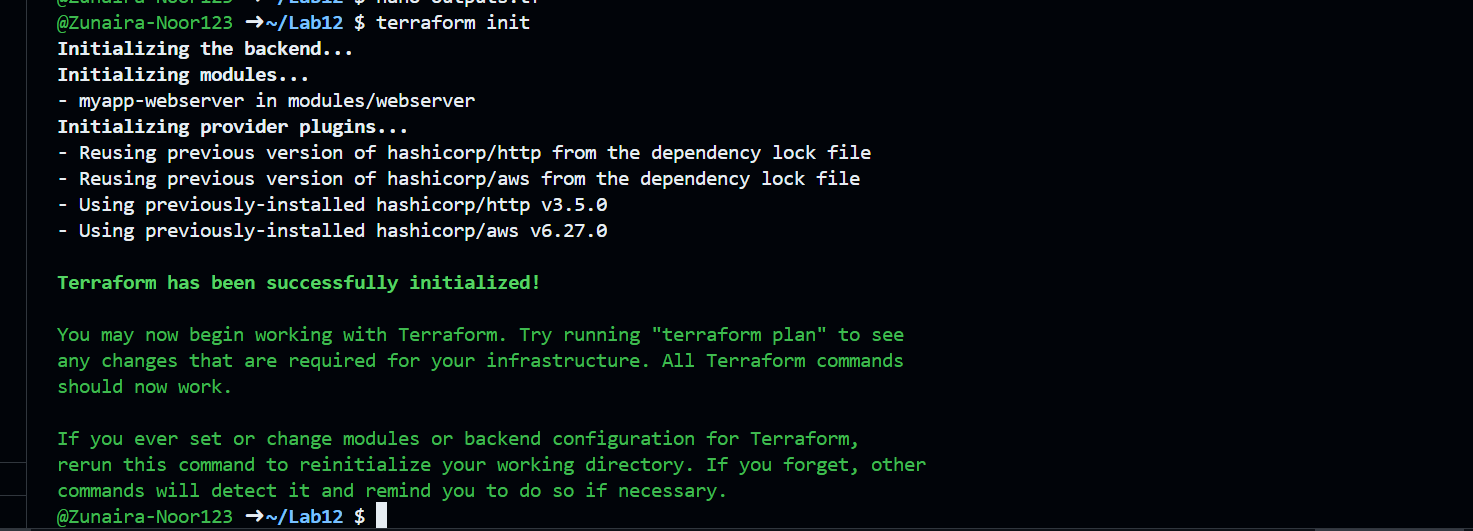
**Step 5: Modify Root main.tf**

****

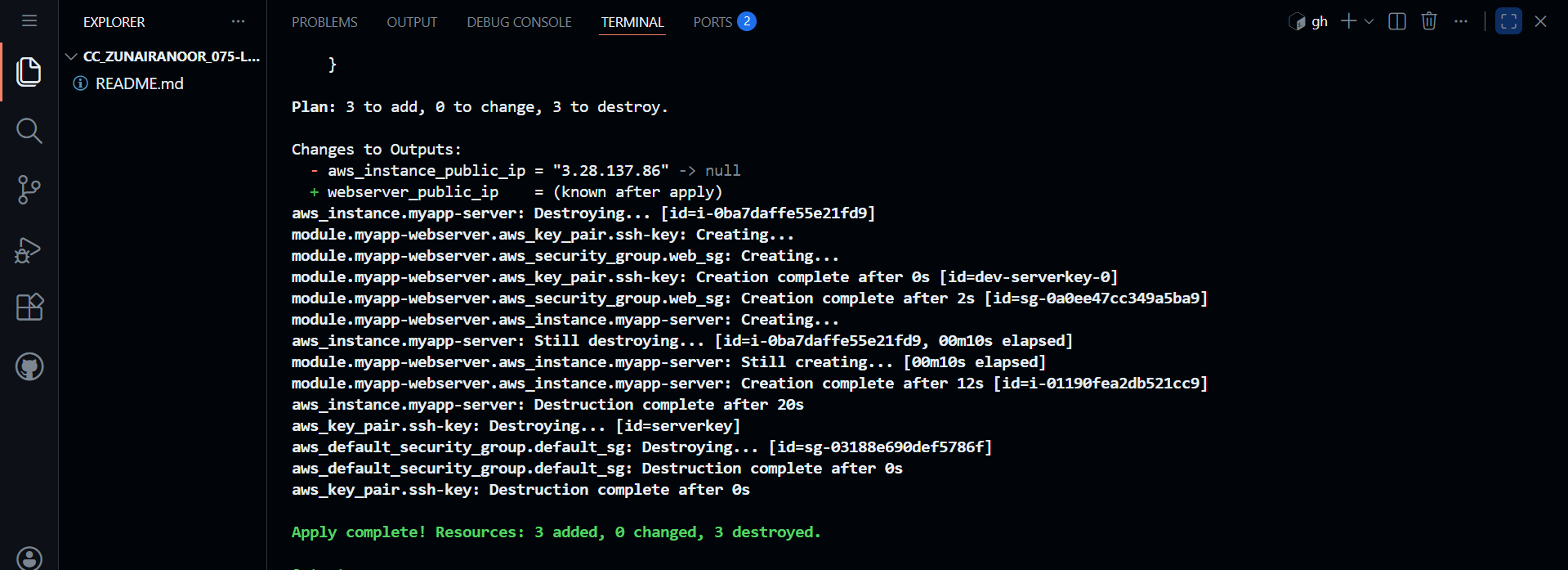
**Step 6: Update outputs.tf**

****

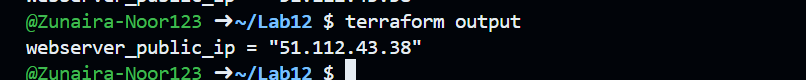
**Step 7: Initialize Terraform**

****

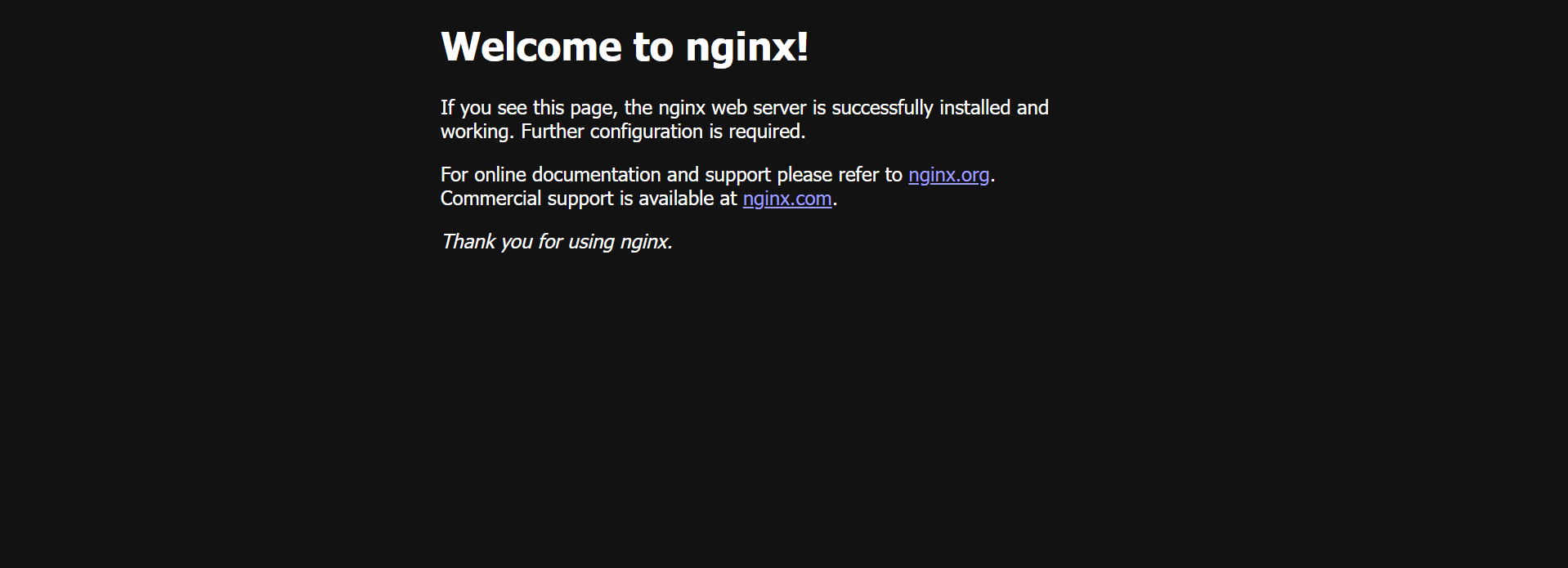
**Step 8: Apply Configuration**

****

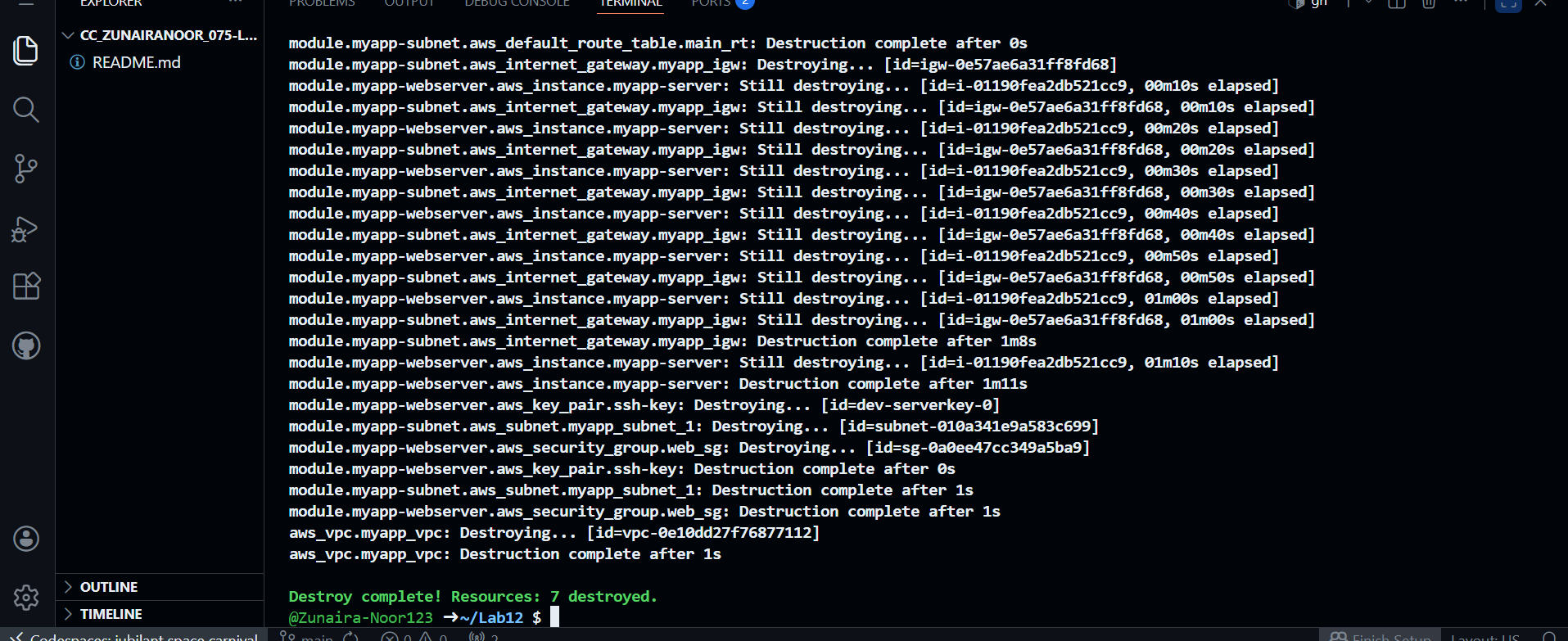
**Step 9: Display Output**

****

**Step 10: Test Nginx**

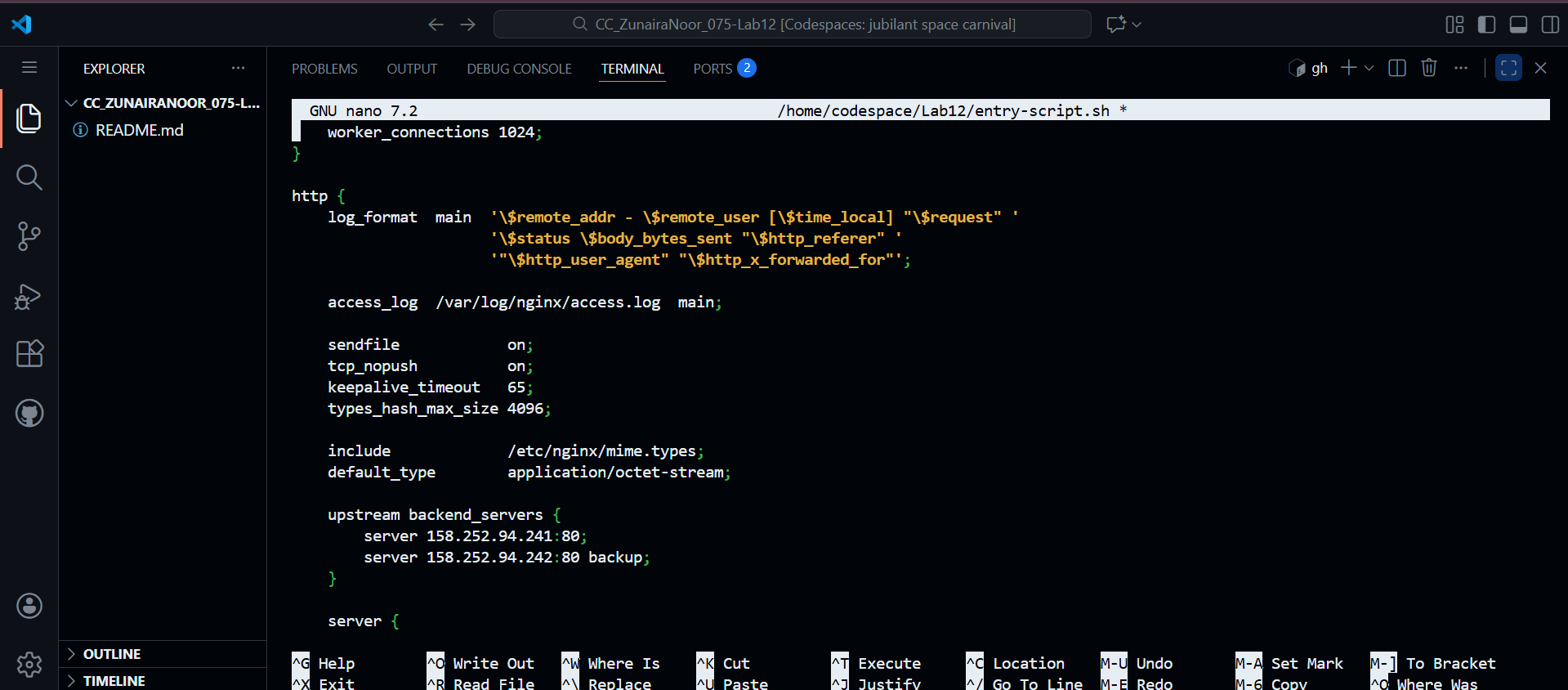
****

**Step 11: Destroy Resources**

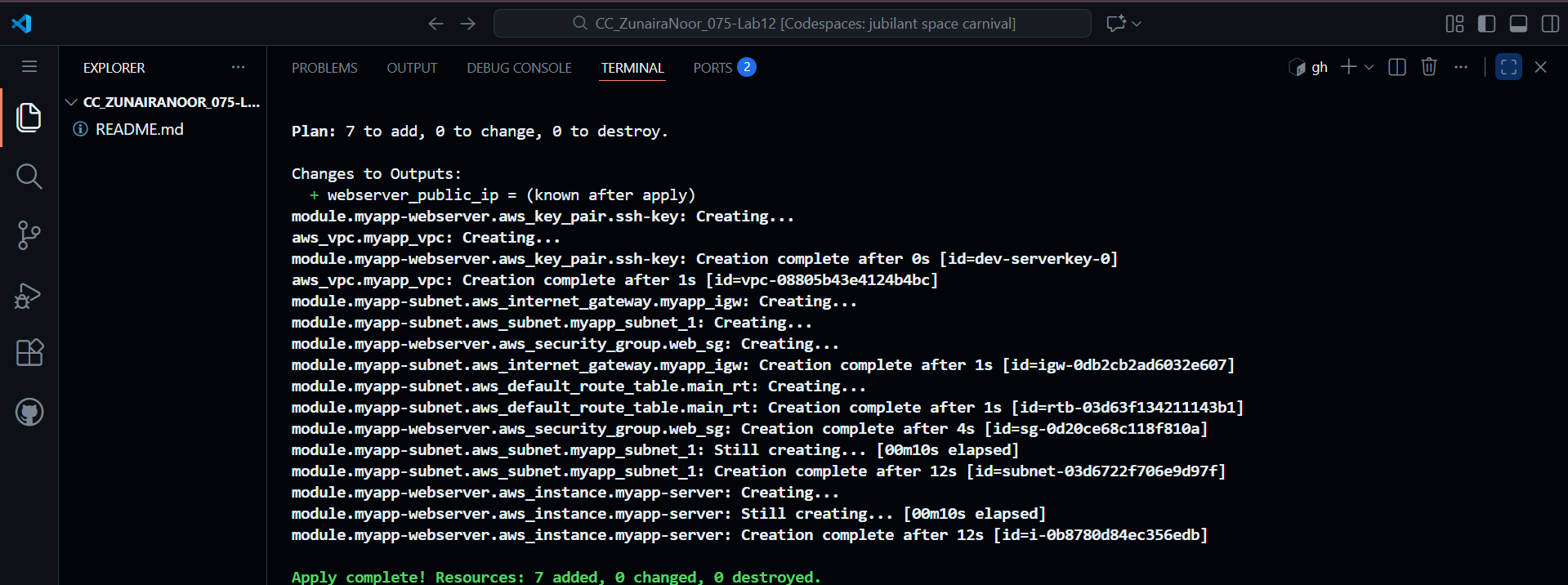
****

**Task 6 — HTTPS with Self-Signed Certificates**

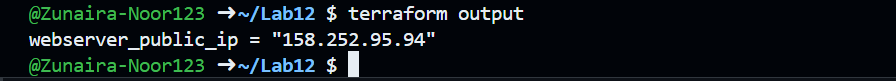
**Step 1: Update entry-script.sh**

****

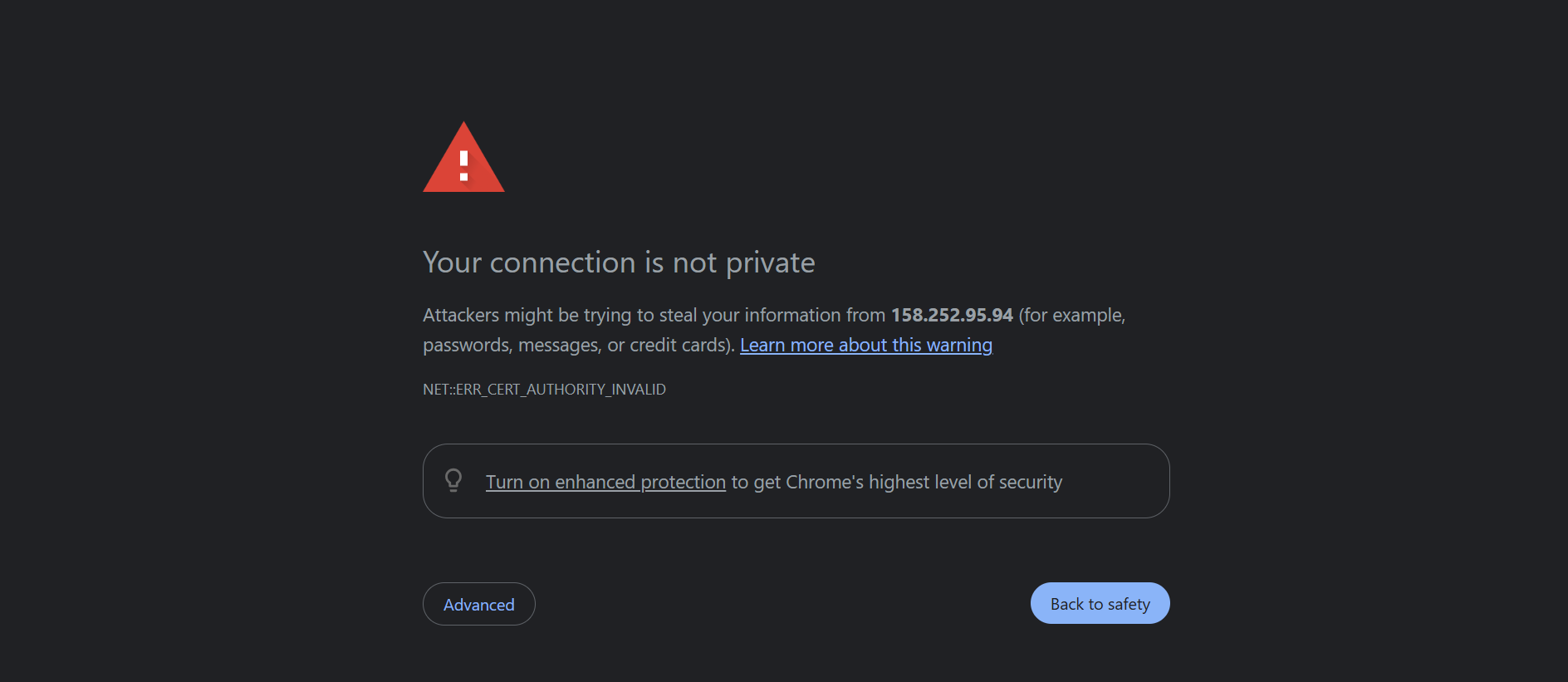
**Step 2: Apply Terraform**

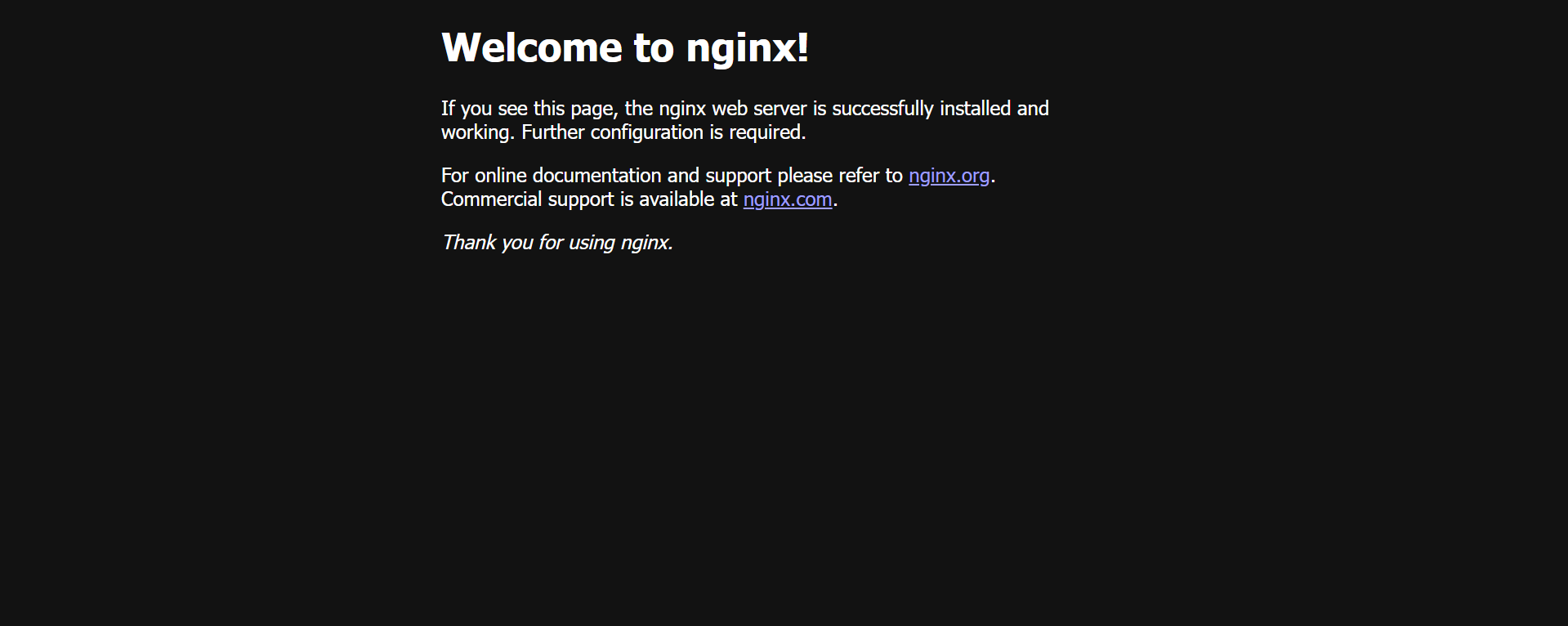
****

**Step 3: Display Output**

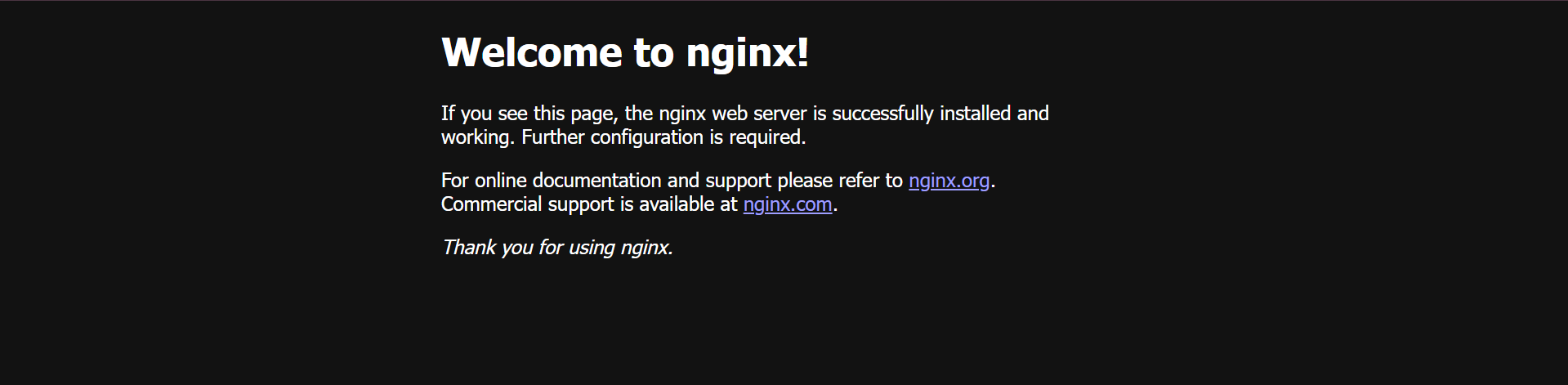
****

**Step 4: Test HTTPS in Browser**

****

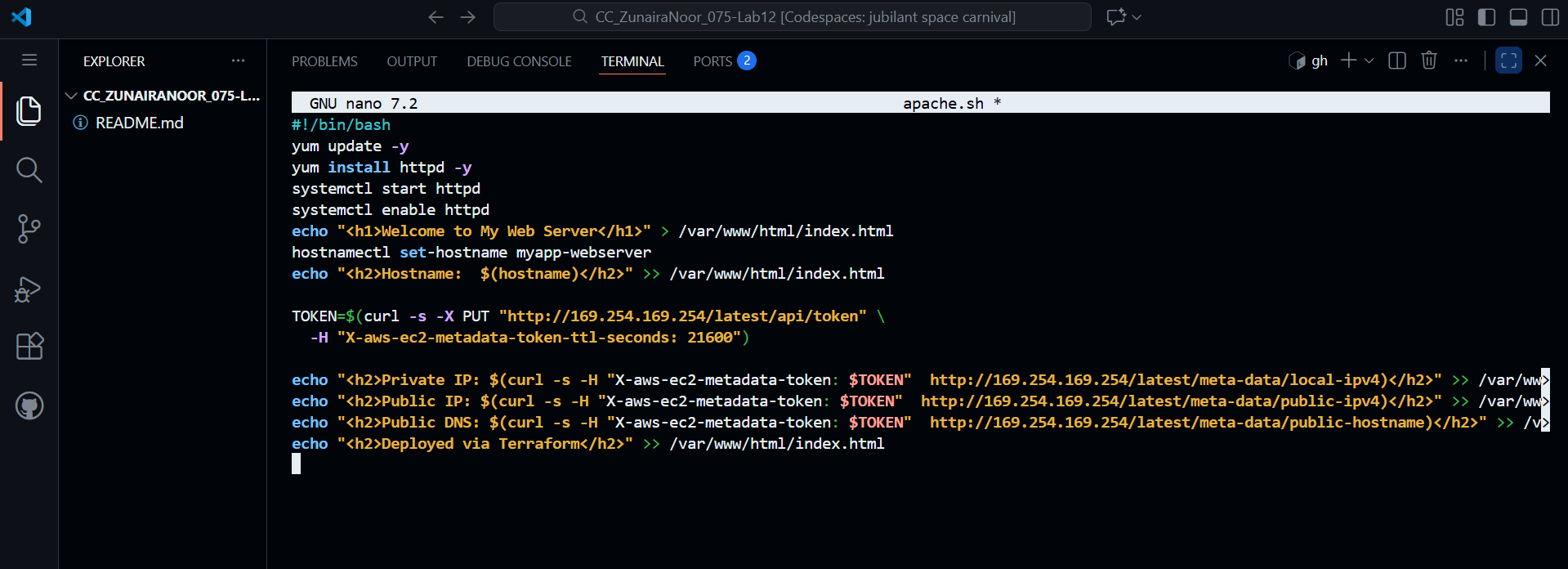
****

**Step 5: Verify HTTP to HTTPS Redirect**

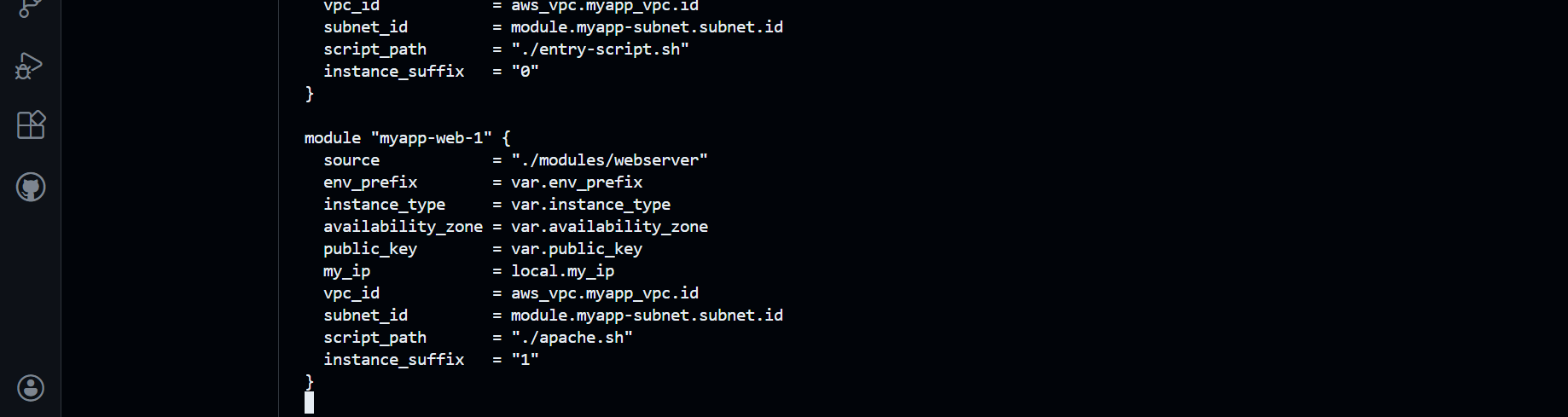
****

**Task 7 — Nginx as Reverse Proxy**

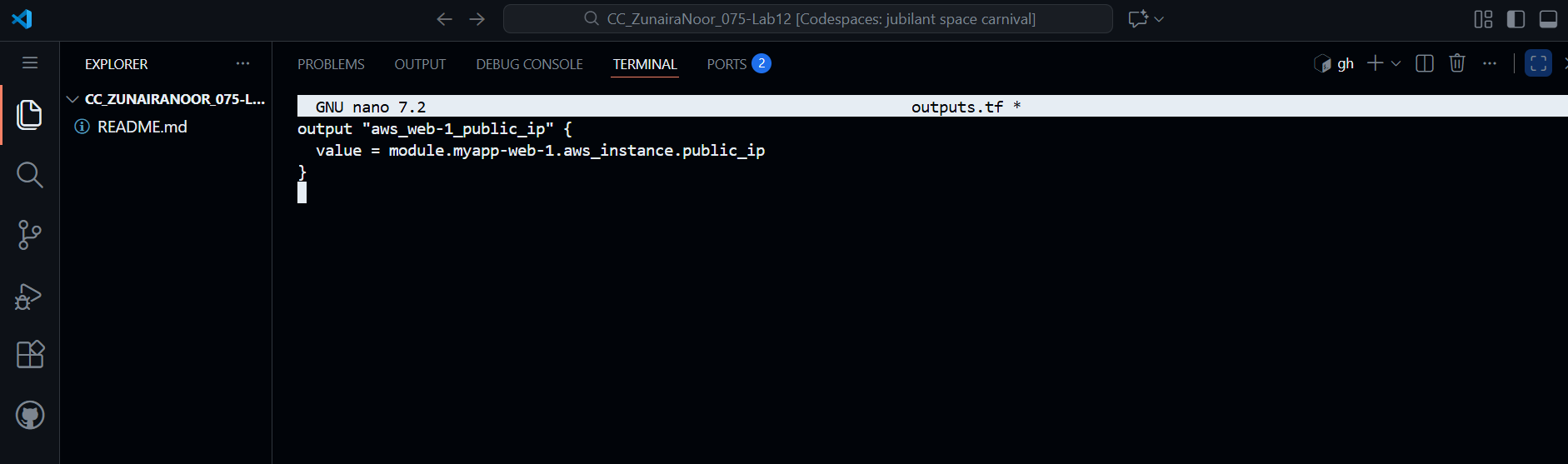
**Step 1: Create apache.sh Script (Backend Web Server)**

****

**Step 2: Add Backend Module myapp-web-1 to main.tf**

****

**Step 3: Update outputs.tf**

****

**Step 4: Apply Terraform**

****

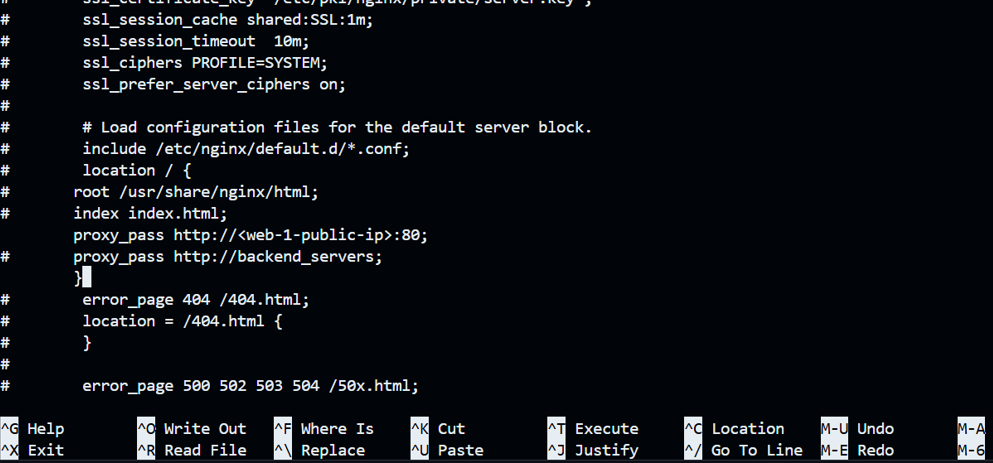
**Step 5: Get Outputs**

****

**Step 6: SSH into Nginx Webserver**

****

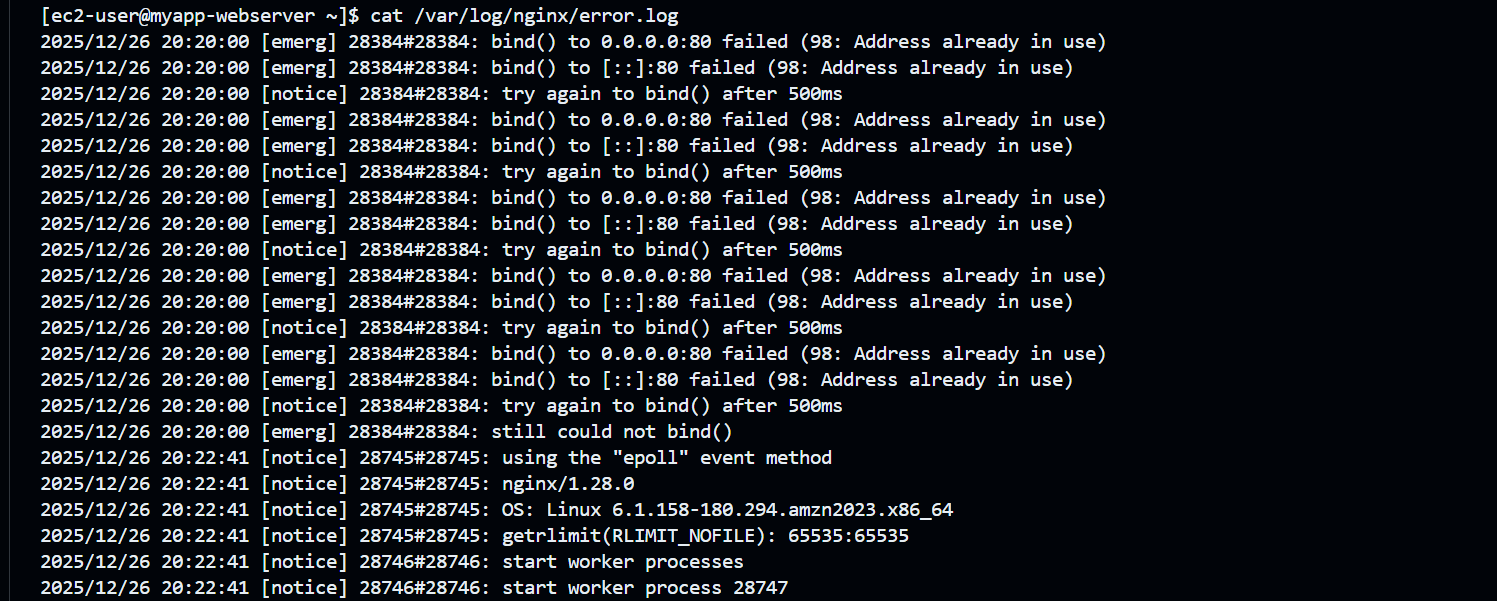
**Step 7: Edit Nginx Config for Reverse Proxy**

****

**Step 8: Restart Nginx**

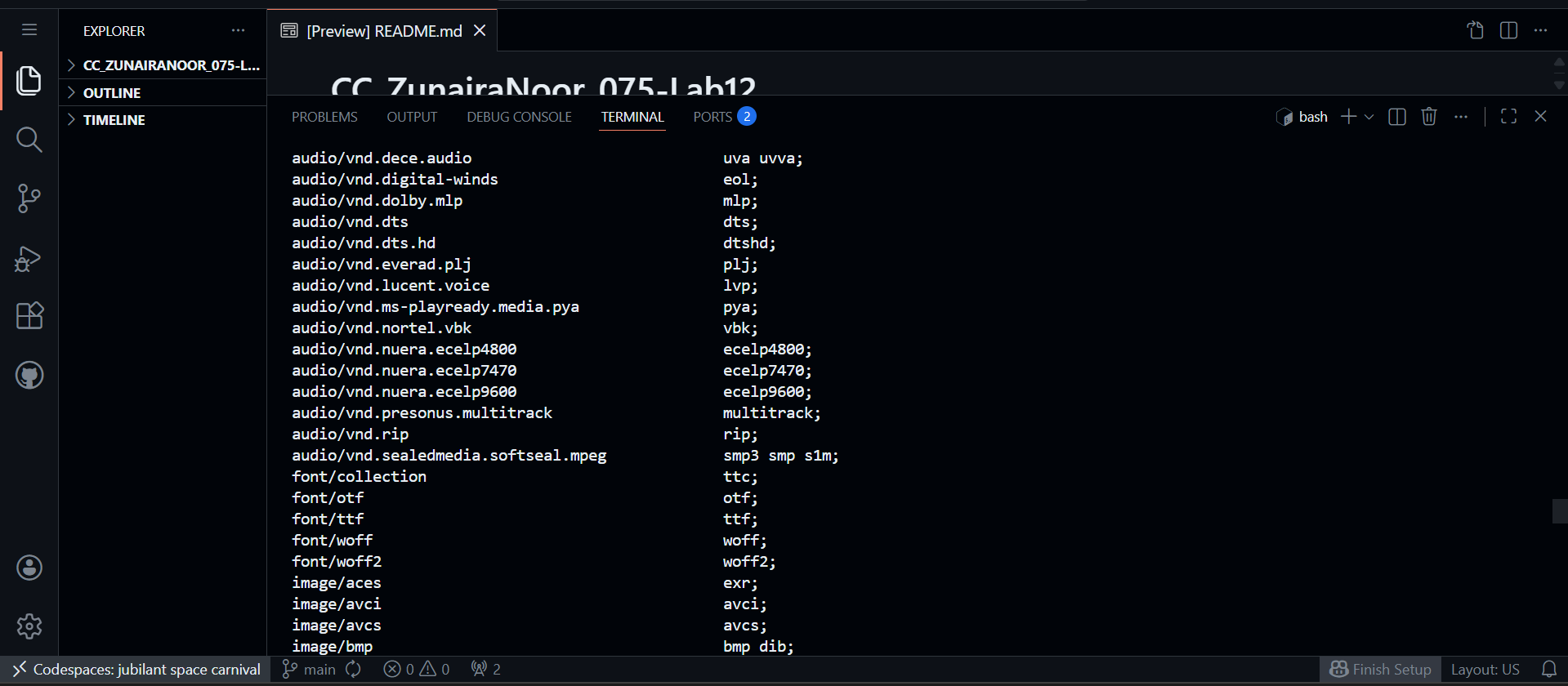
****

**Step 9: View Nginx Logs and Files**

****

**Step 10: Test Reverse Proxy in Browser**

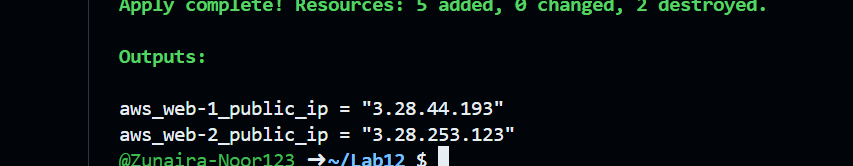
****

****

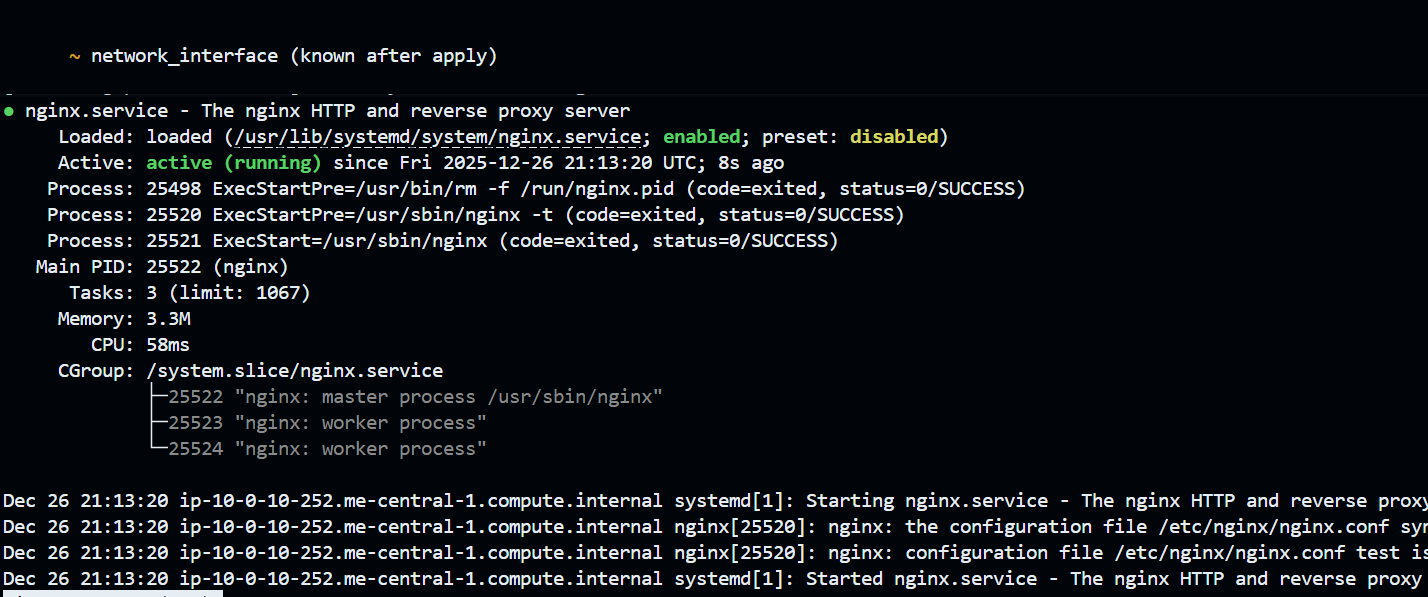
****

**Task 8 — Configure Nginx as load balancer**

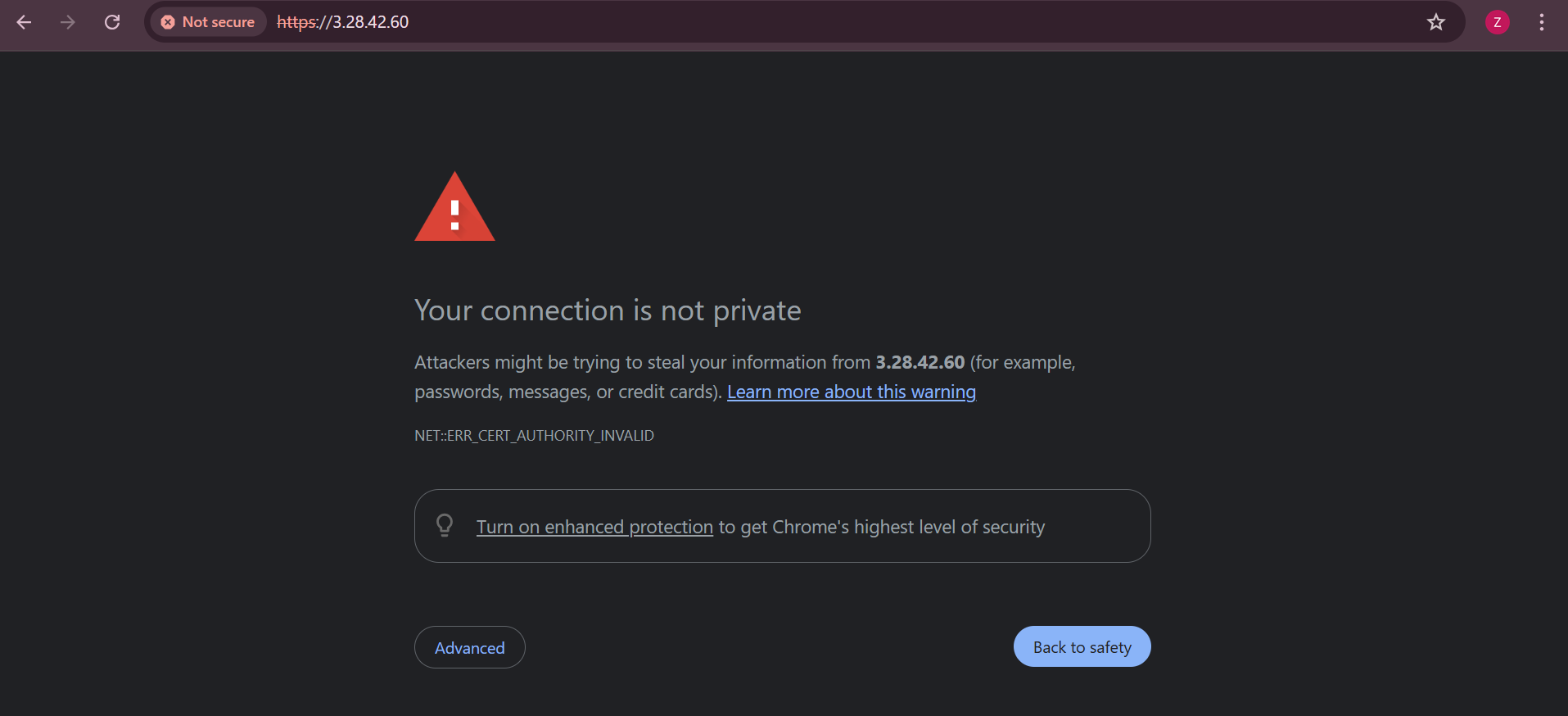
**Apply Terraform**

****

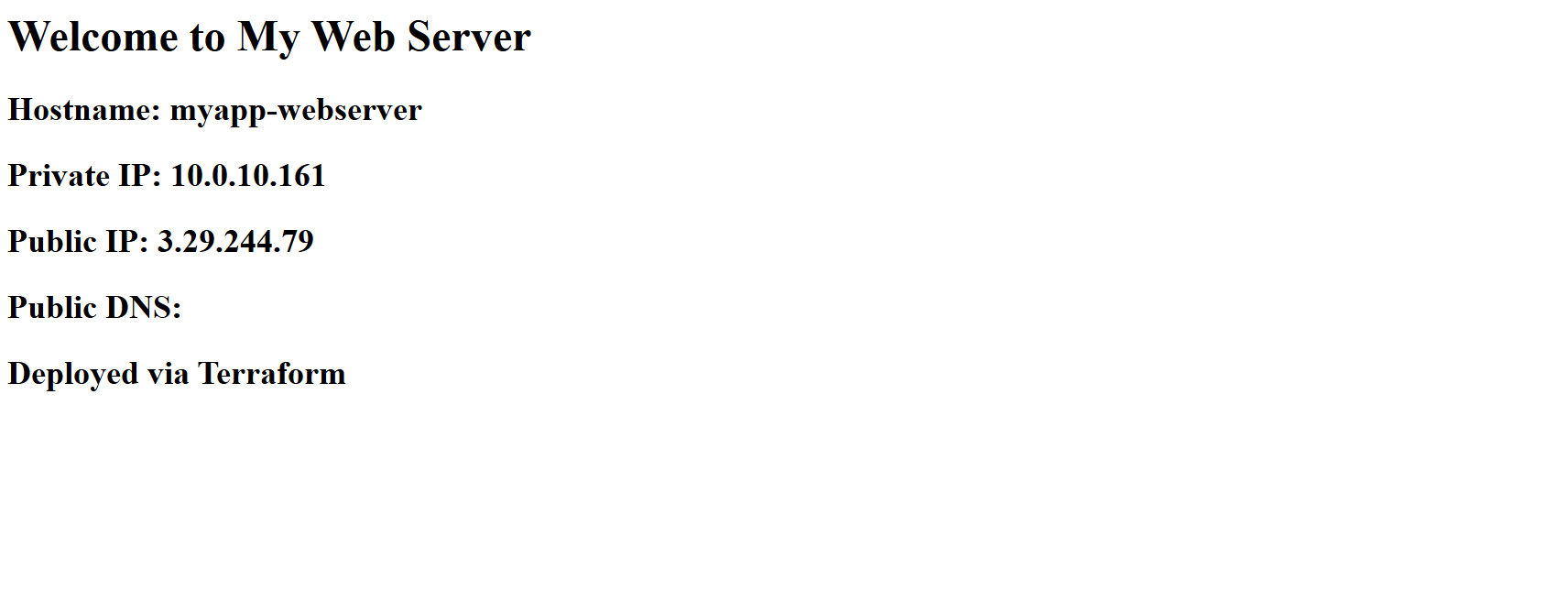
**Restart Nginx**

****

**Test load balancing**

****

**Web1**

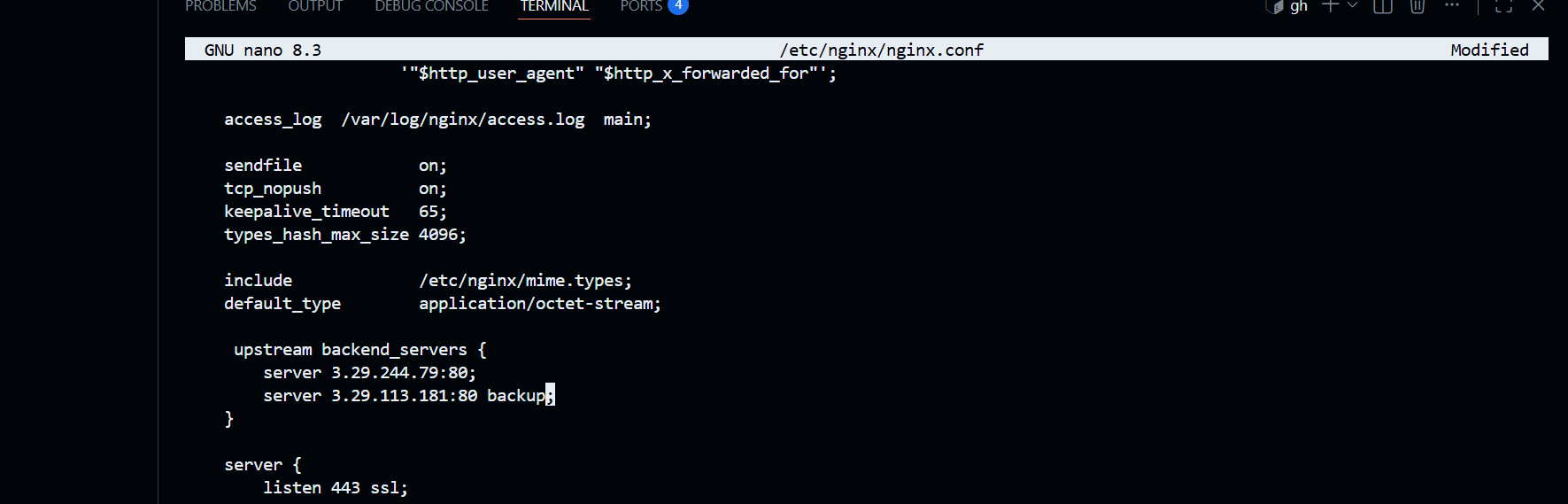
****

**Web2**

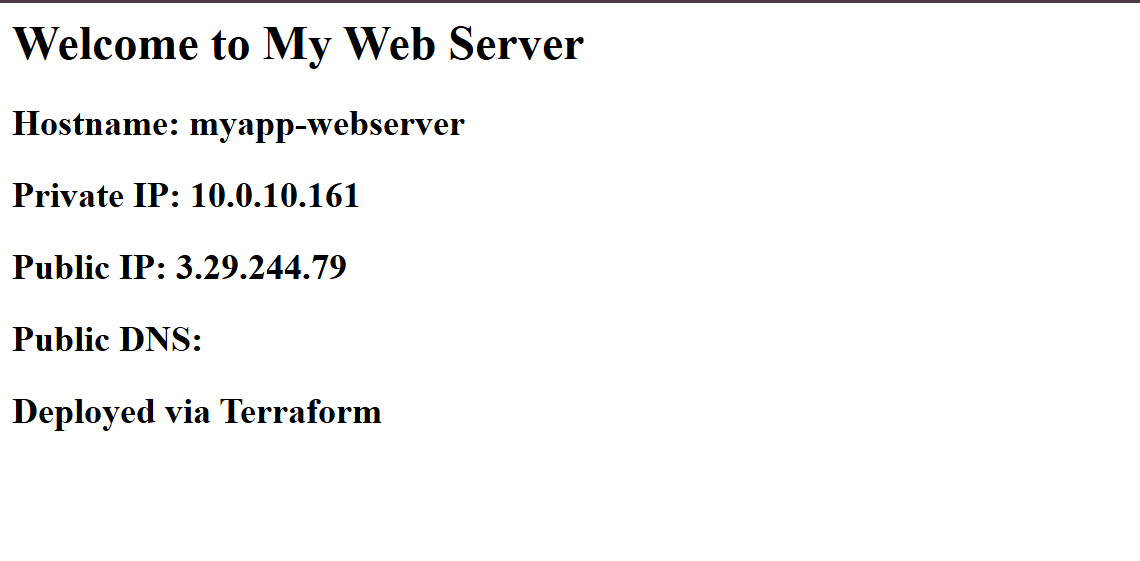
****

**Task 9 — Configure high availability with backup servers**

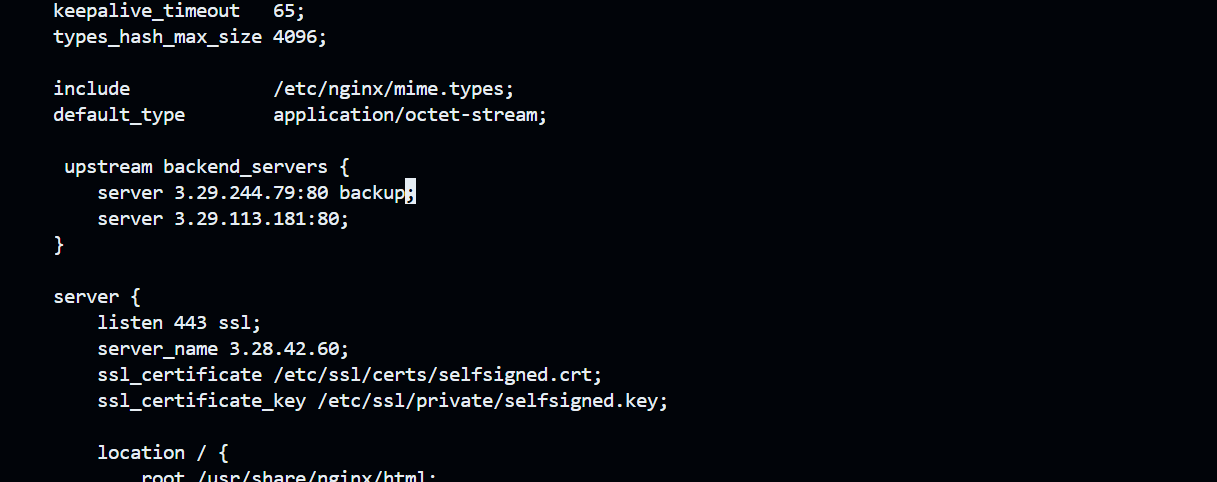
**Edit Nginx configuration for high availability:**

****

**Test in browser**

****

**Switch backup configuration:**

****

**Restart Nginx:**

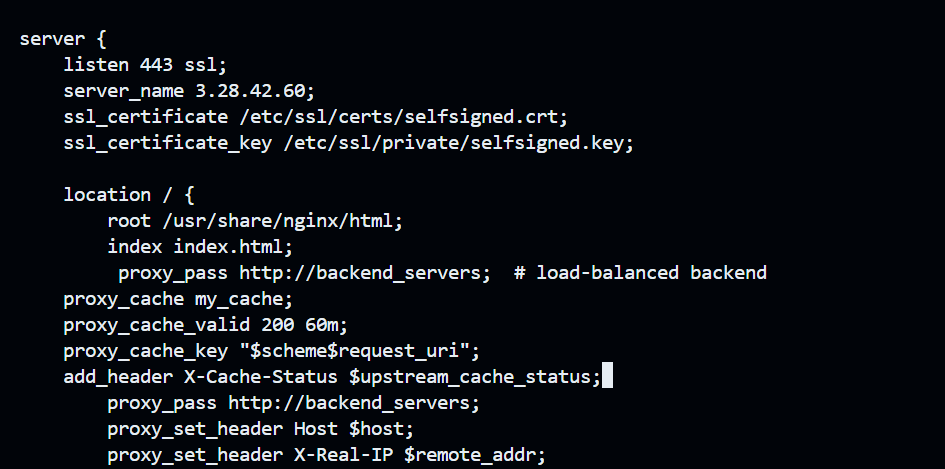
****

**Test in browser:**

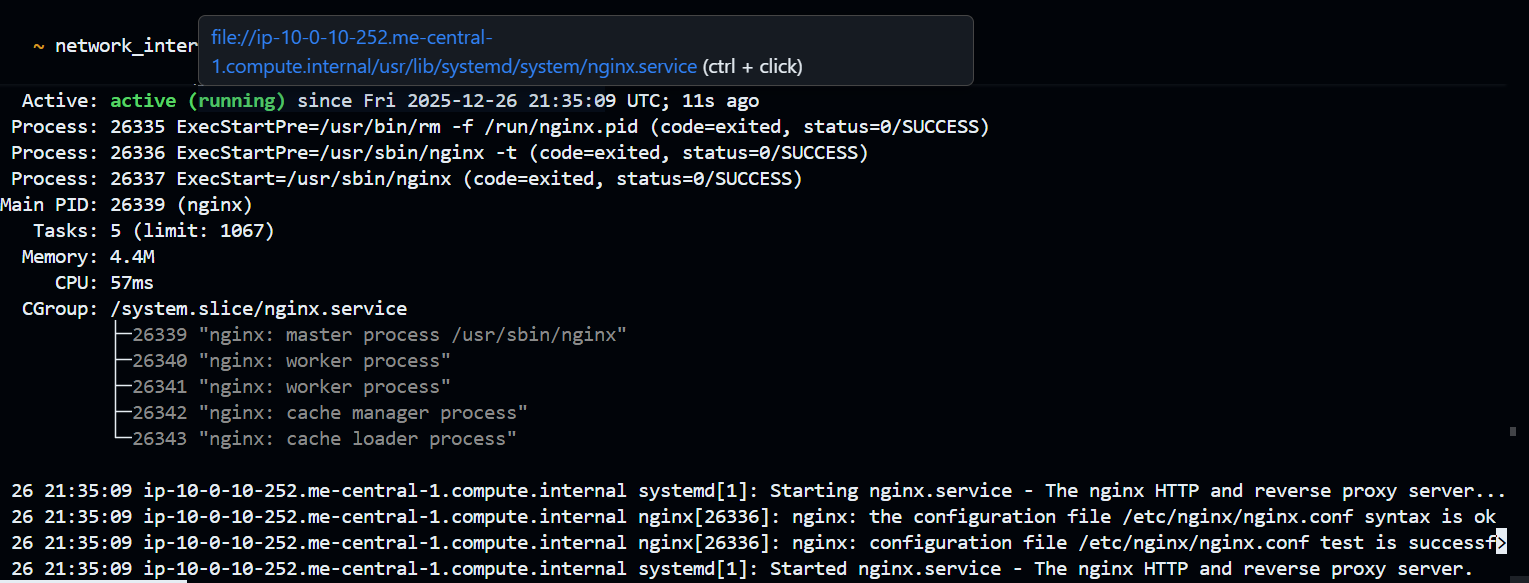
****

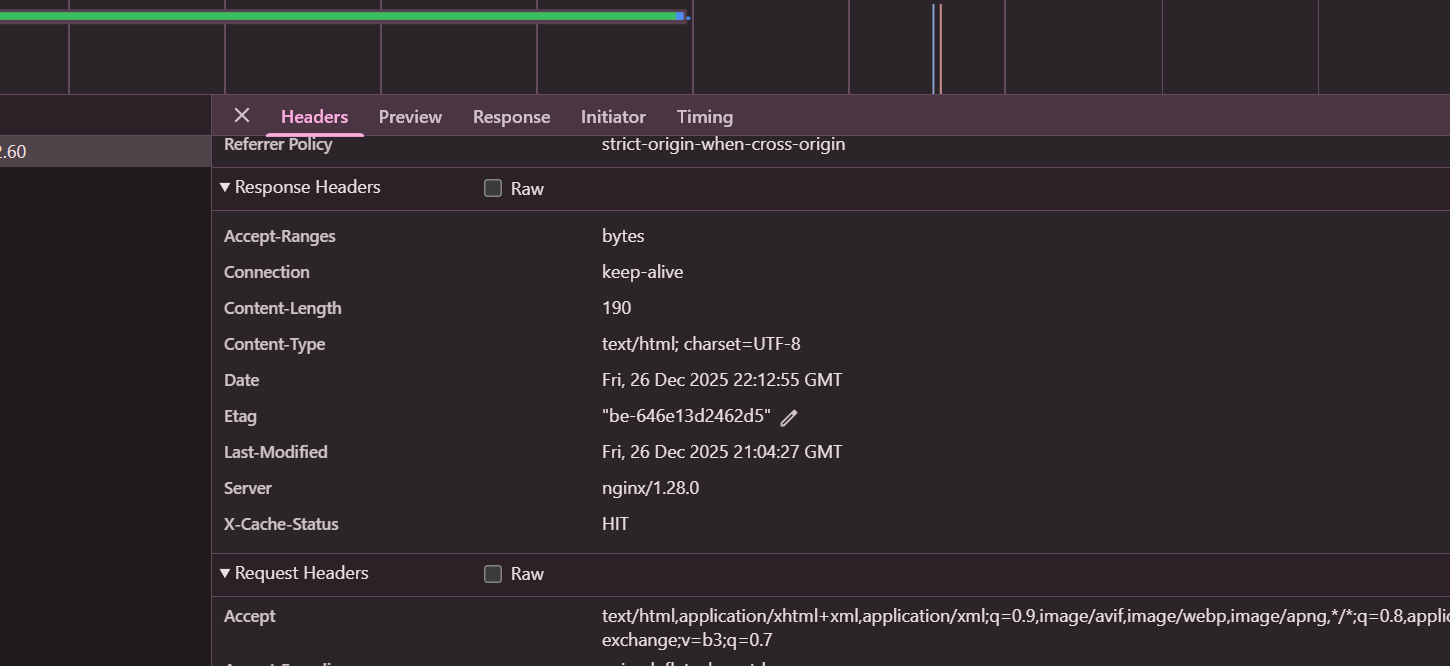
**Task 10 — Enable Nginx caching**

**Add proxy cache configuration in the http block and location block:**

****

1. **Restart Nginx:**

****

****