**CLOUD COMPUTING LAB**



**Submitted By: Zunaira Khatoon**

**Submitted To: Engr. Muhammad Shoaib**

**Roll No: BSE-2023-074**

**Section: V-B**

***LAB 11***

Contents

[**Task 0 - Lab Setup (Codespace & GH CLI)** 2](#_Toc219406900)

[**Task 1 — Provider & Basic variable (variable precedence)** 2](#_Toc219406901)

[**Task 2 — Variable validation & sensitive / ephemeral variables** 2](#_Toc219406902)

[**Task 3 — Project-level variables, locals, and outputs** 2](#_Toc219406903)

[**Task 4 — Maps and Objects** 2](#_Toc219406904)

[**Task 5 — Collections: list, tuple, set & mutation via locals** 3](#_Toc219406905)

[**Task 6 — Null, any type & dynamic values** 3](#_Toc219406906)

[**Task 7 — Git ignore** 3](#_Toc219406907)

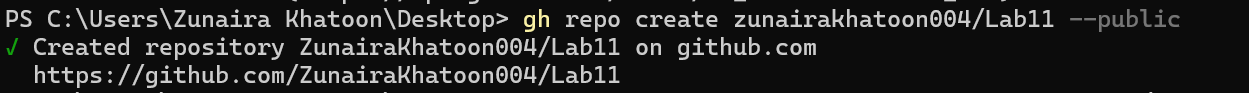
[**Task 8 — Clean-up then build real infra (VPC, Subnet, IGW, routing, default route table)** 3](#_Toc219406908)

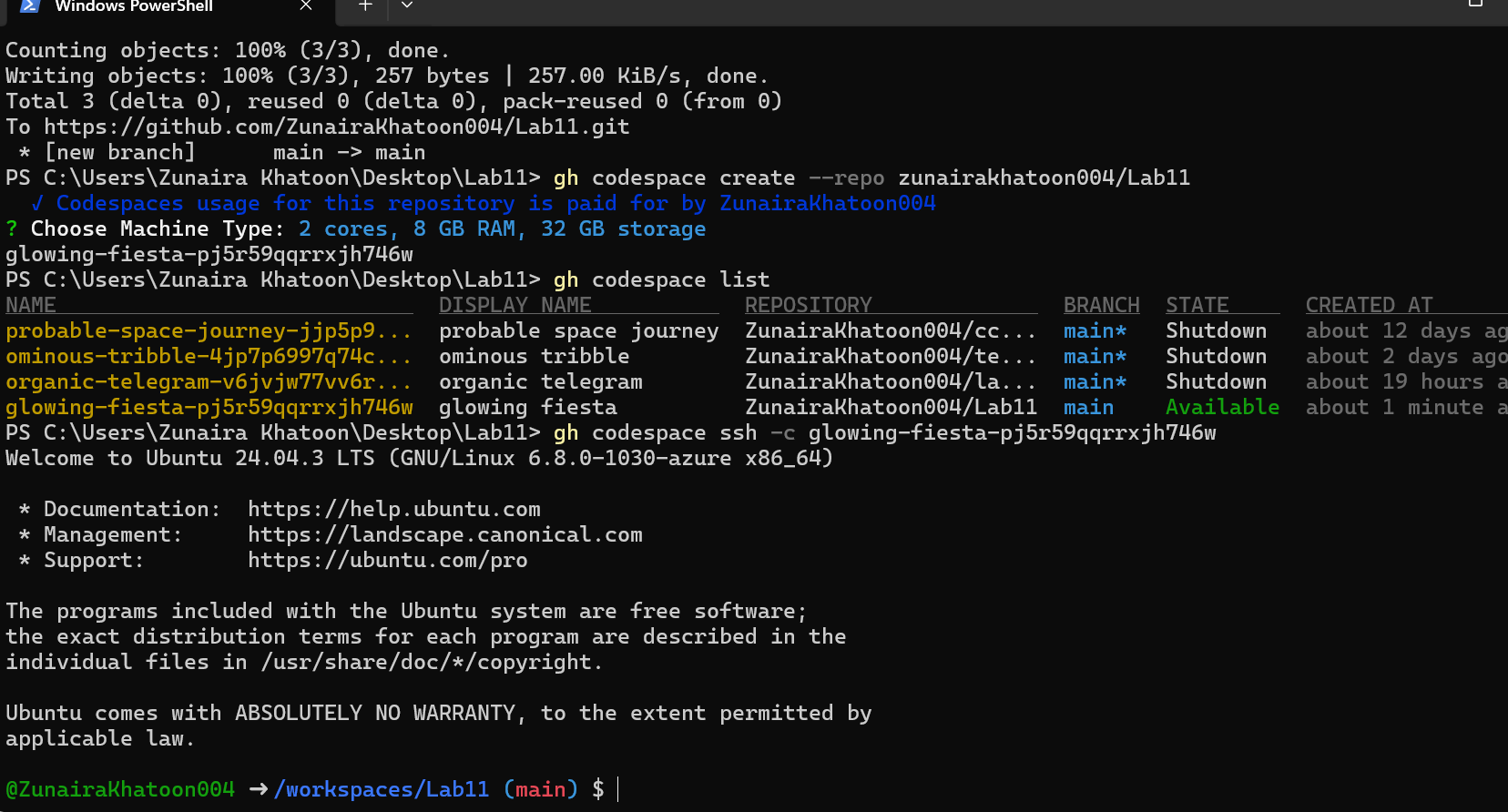
[**Task 9 — Security Group, key pair, EC2 instance, user\_data & nginx** 3](#_Toc219406909)

[**Cleanup — Destroy resources & verify state** 4](#_Toc219406910)

# [**Task 0 - Lab Setup (Codespace & GH CLI)**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-0-lab-setup-codespace--gh-cli)

* Create Codespace & connect:



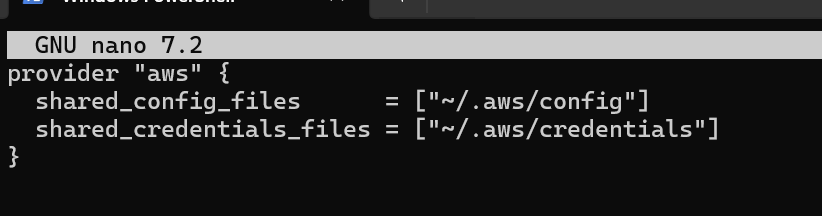


# [**Task 1 — Provider & Basic variable (variable precedence)**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-1--provider--basic-variable-variable-precedence)

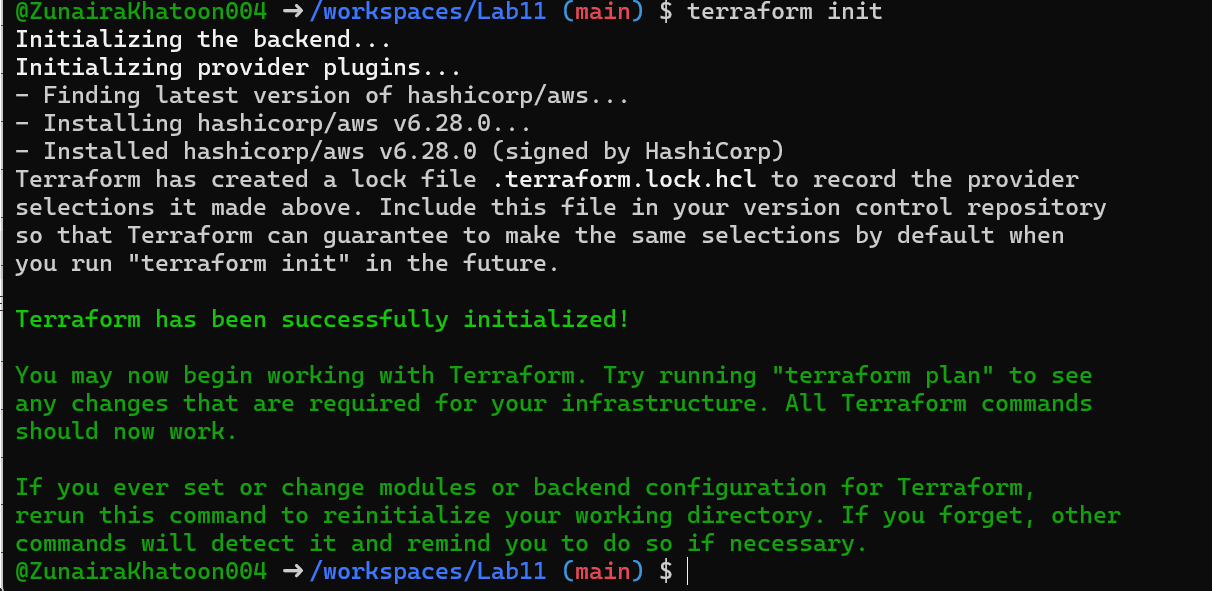
1. In Codespace create main.tf:



1. Edit main.tf and add provider:



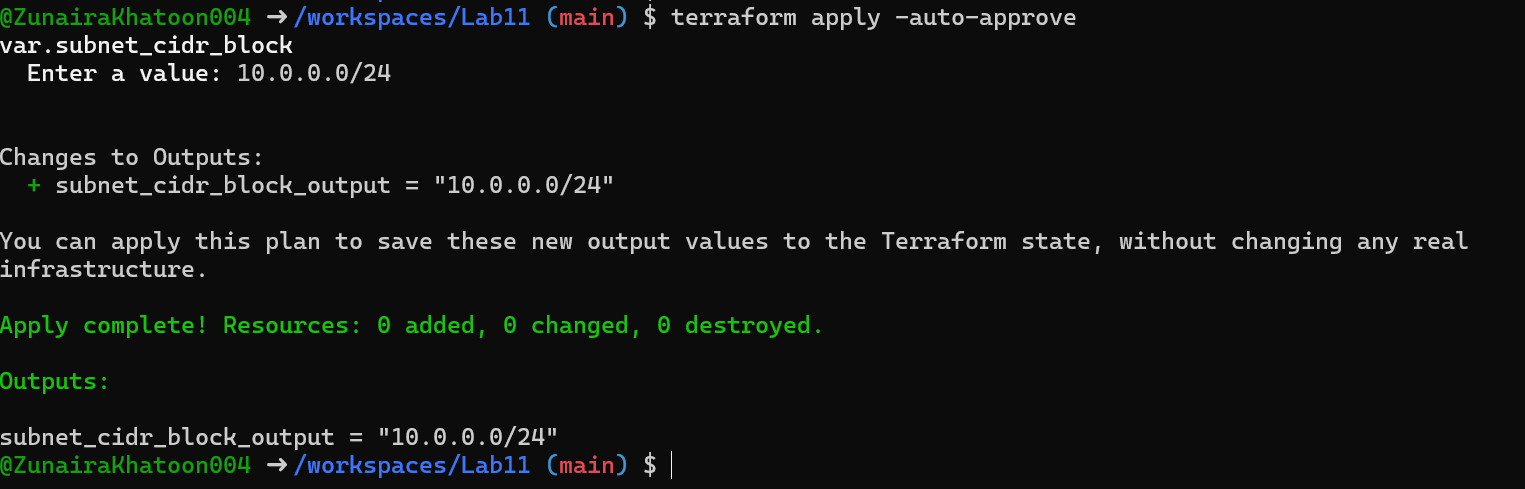
1. Initialize:



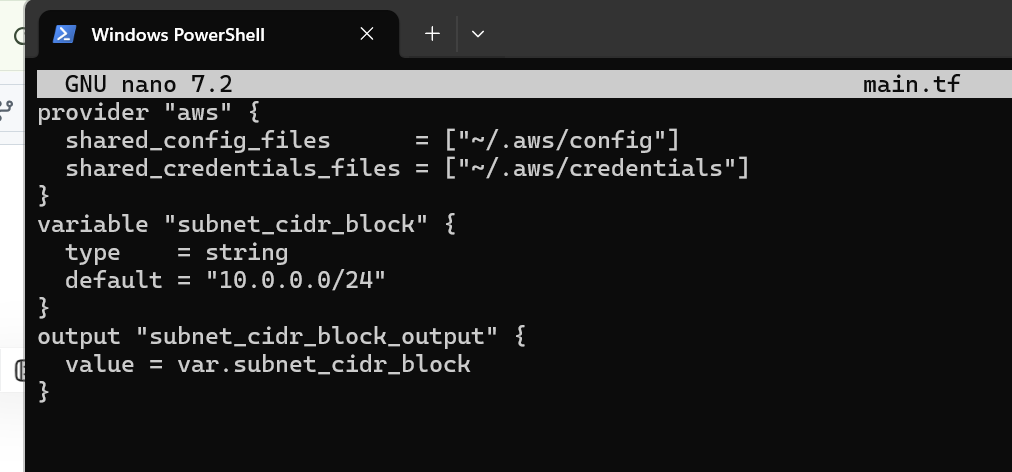
1. Define a variable and output:

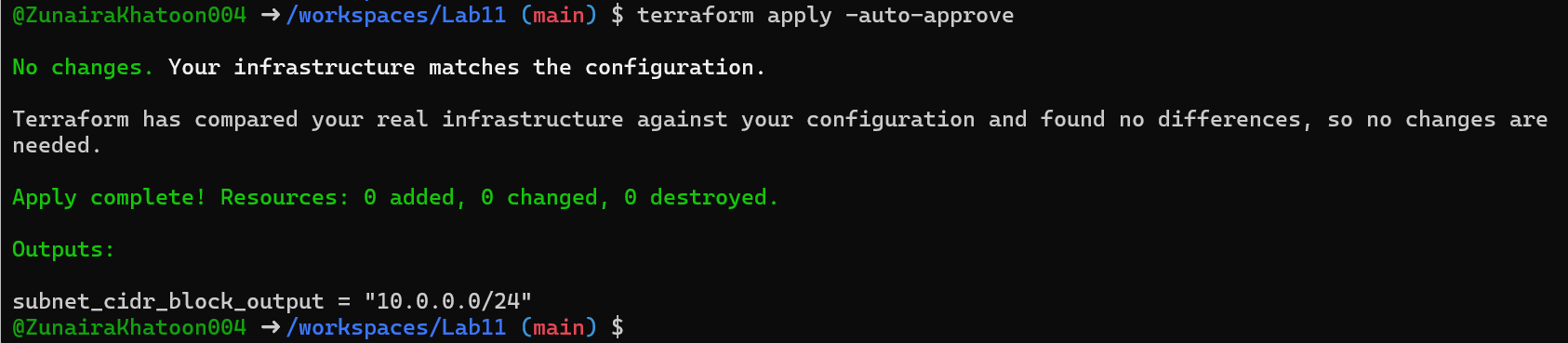


1. Run apply (first time without defaults):

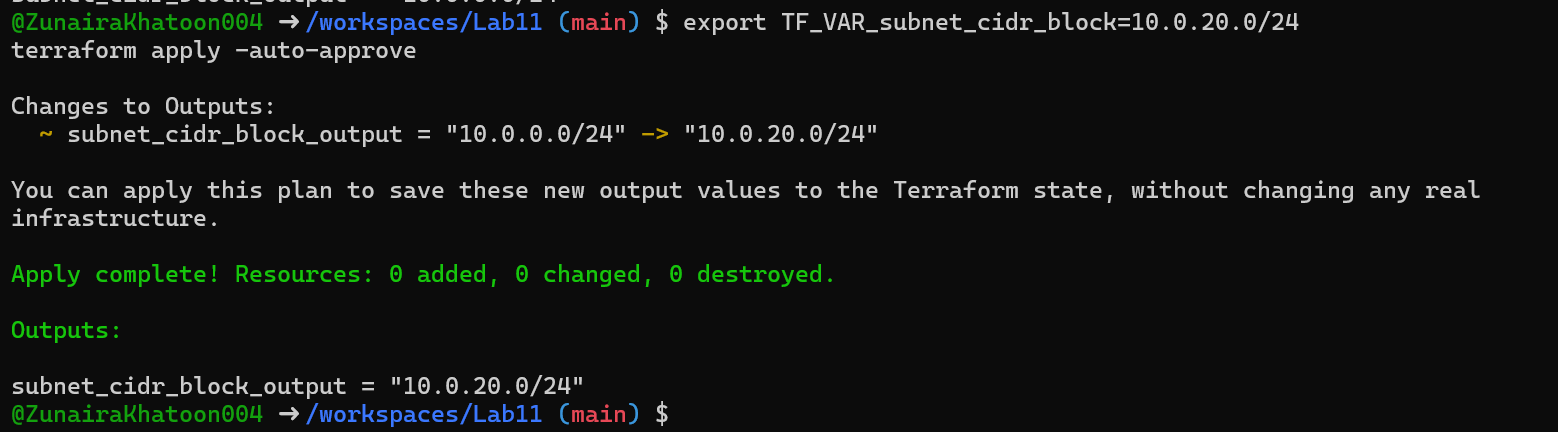


1. Add a default to the variable:

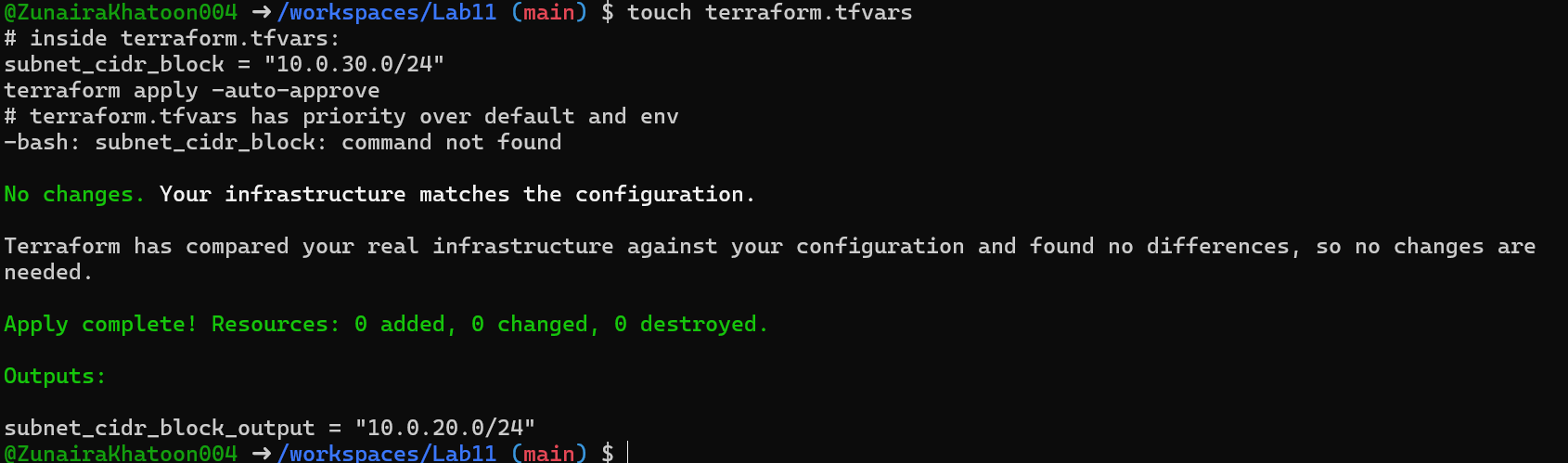




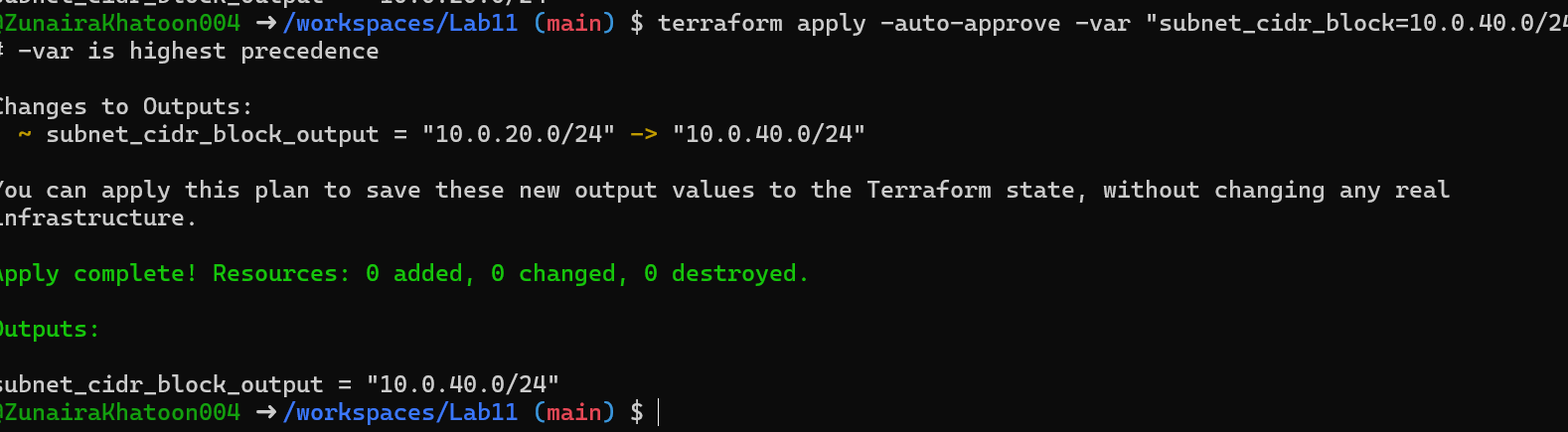
1. Export environment variable in Codespace shell:



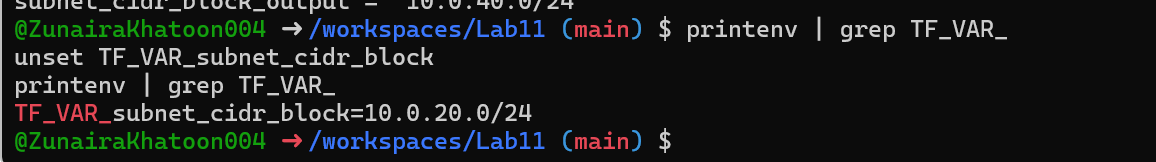
1. Create terraform.tfvars overriding values:



1. Override with -var:

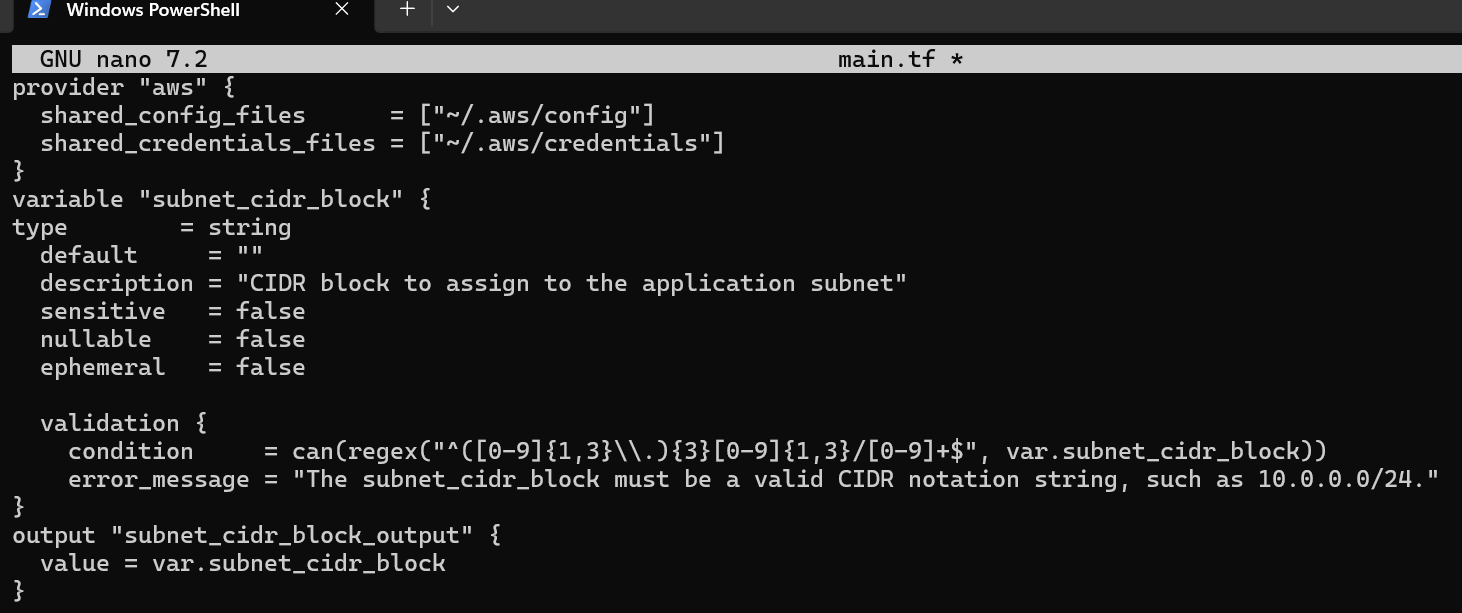


1. Show and unset env var:

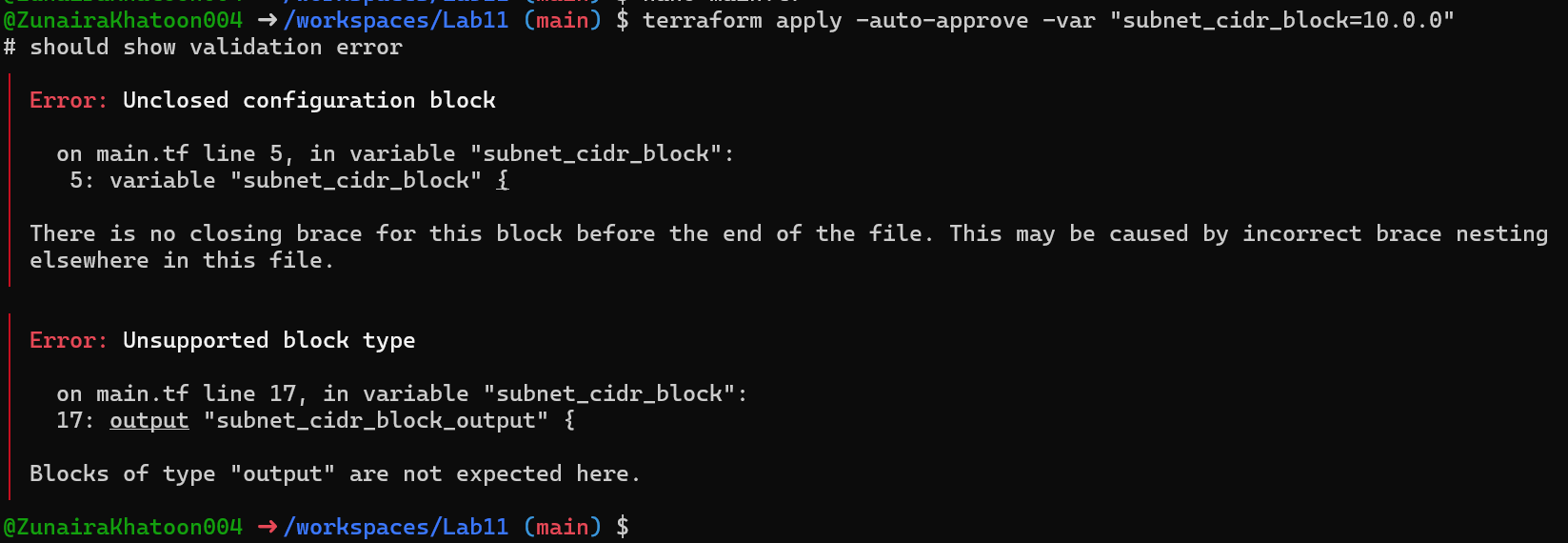


# [**Task 2 — Variable validation & sensitive / ephemeral variables**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-2--variable-validation--sensitive--ephemeral-variables)

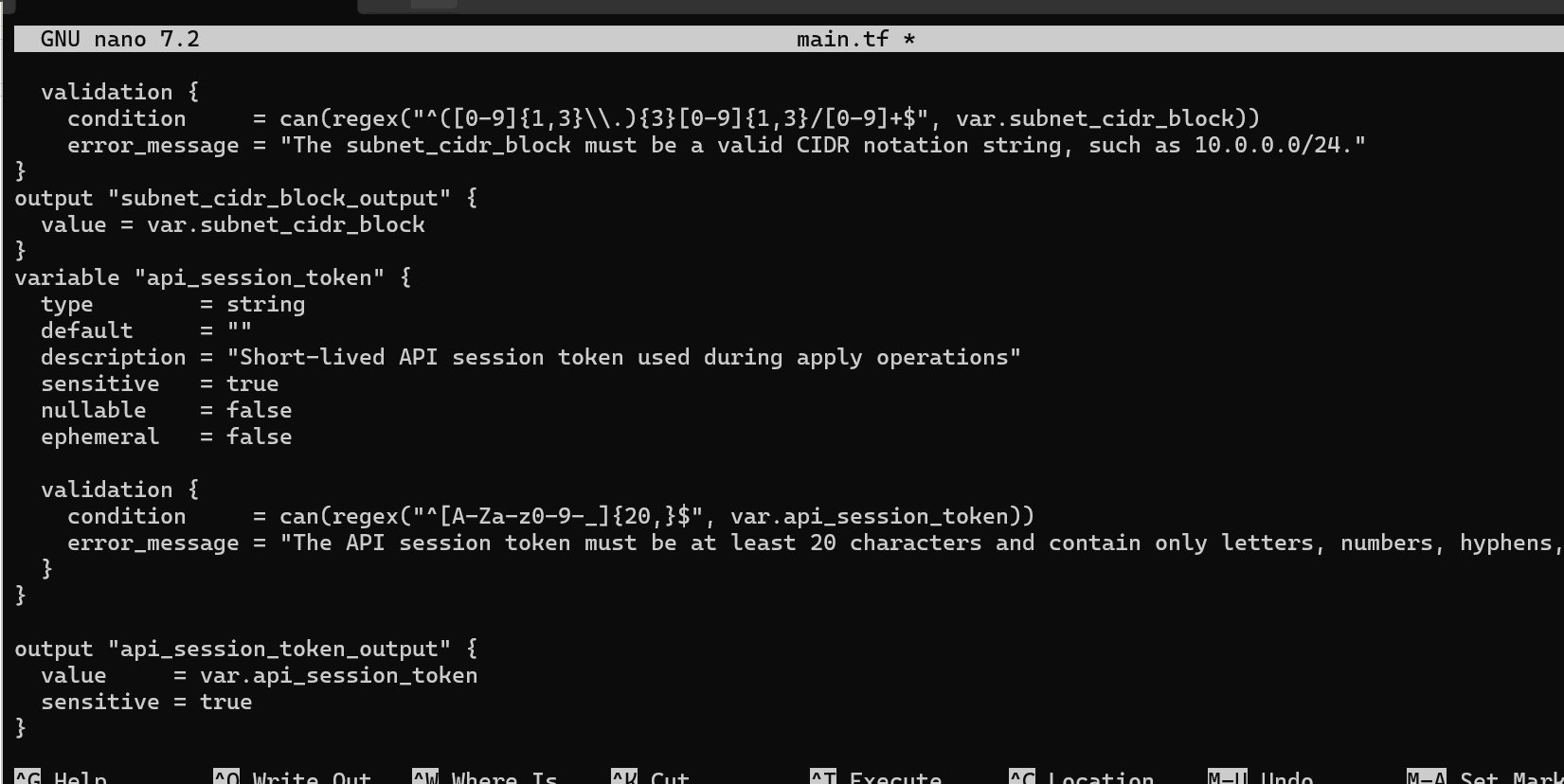
1. Replace subnet\_cidr\_block variable with this (validation included):



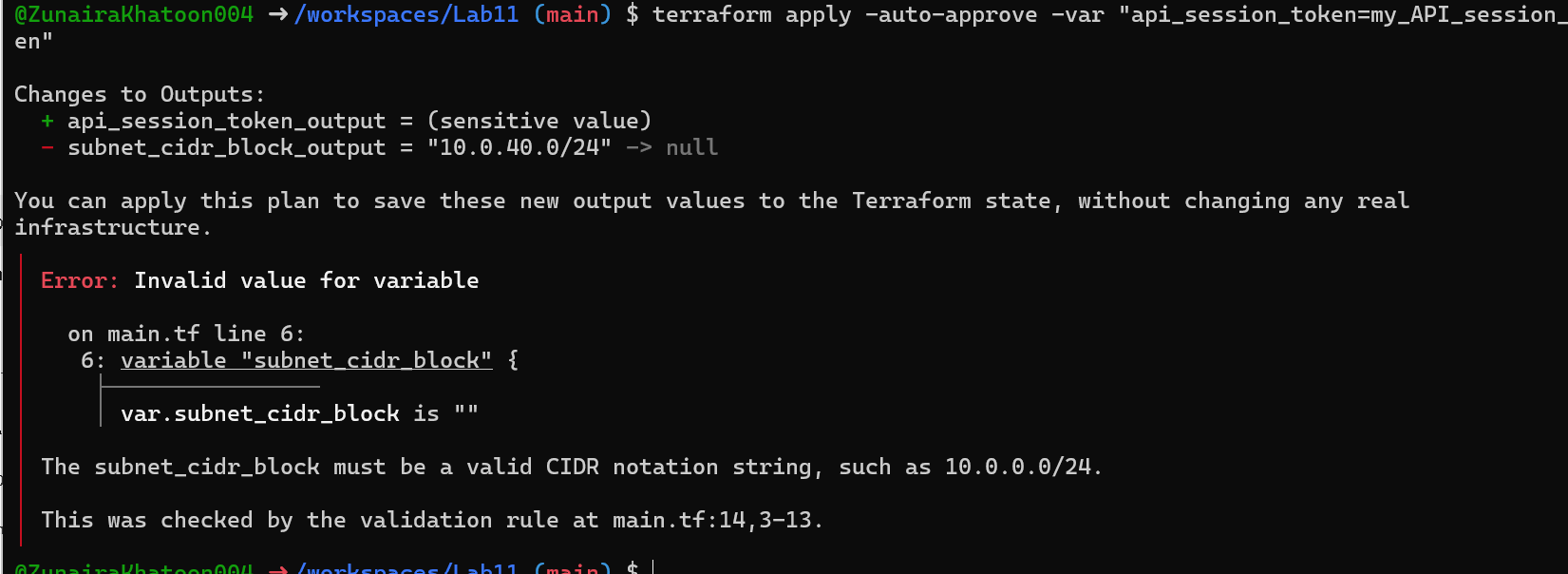
1. Test validation failure:



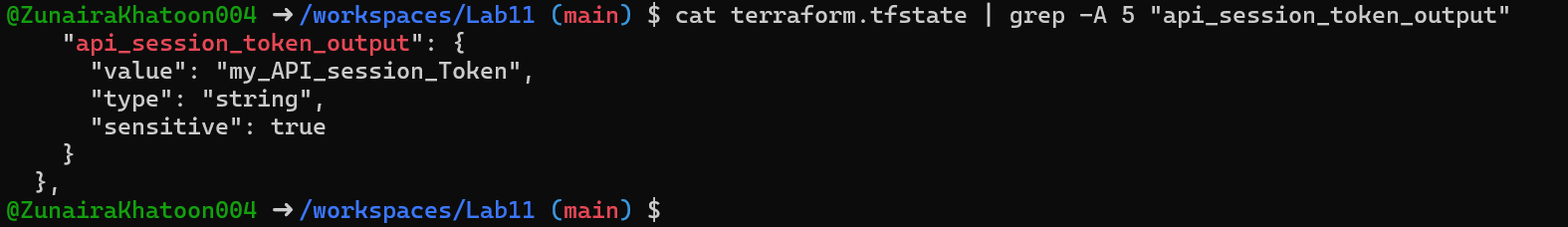
1. Create a sensitive variable api\_session\_token and output (sensitive):



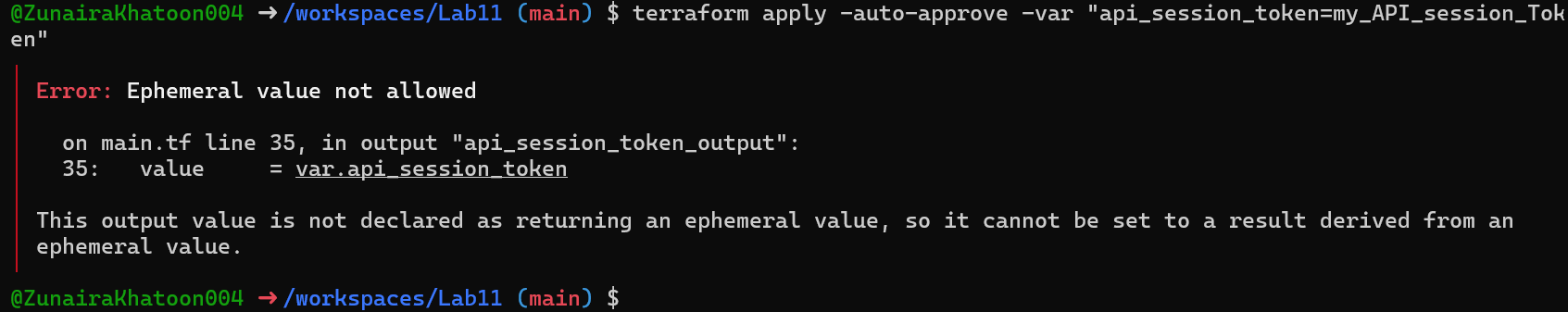
1. Run with -var to observe sensitive output behavior:



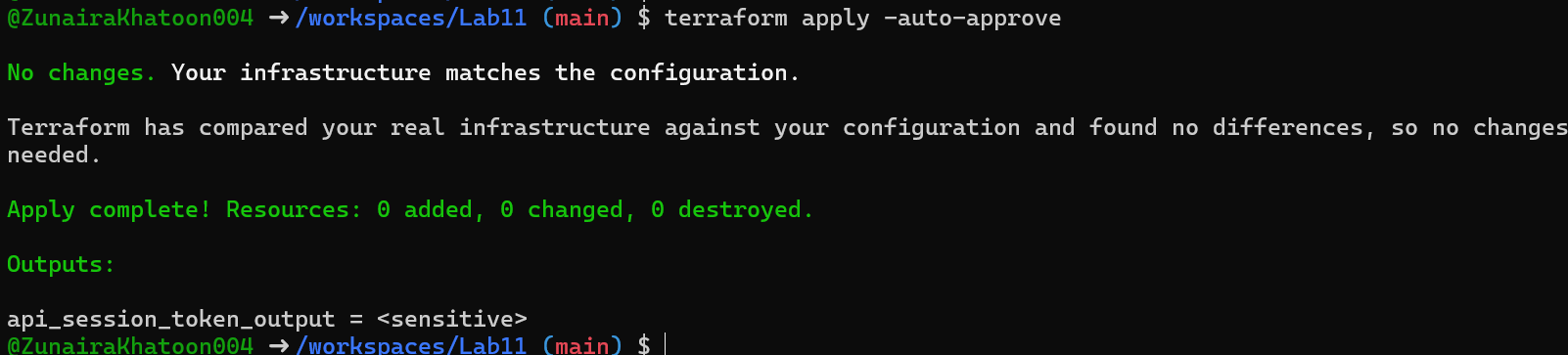
1. Check terraform.state for the sensitive output:



1. Make variable ephemeral to hide from state:

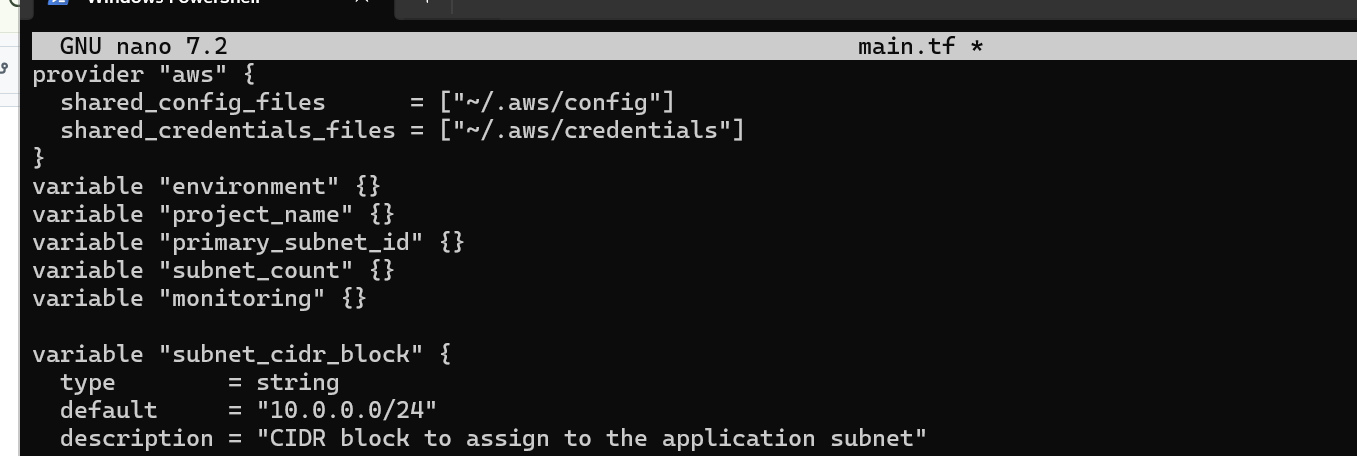


1. Set default to test local default:

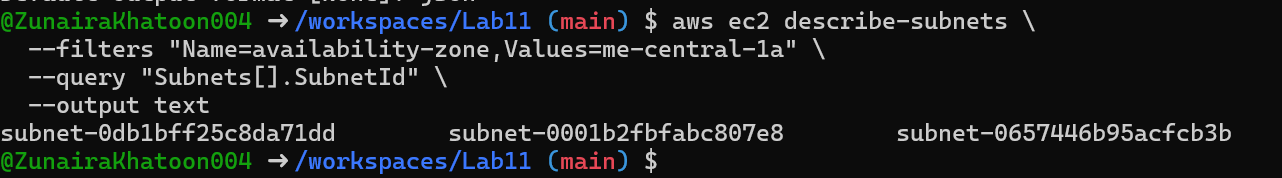


# [**Task 3 — Project-level variables, locals, and outputs**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-3--project-level-variables-locals-and-outputs)

1. Add variables to main.tf:

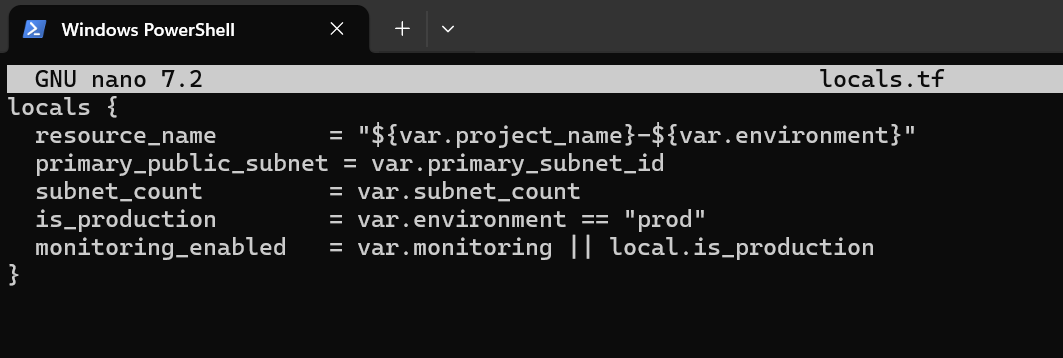


1. Populate terraform.tfvars *after* discovering actual subnet id for availability zone me-central-1a:

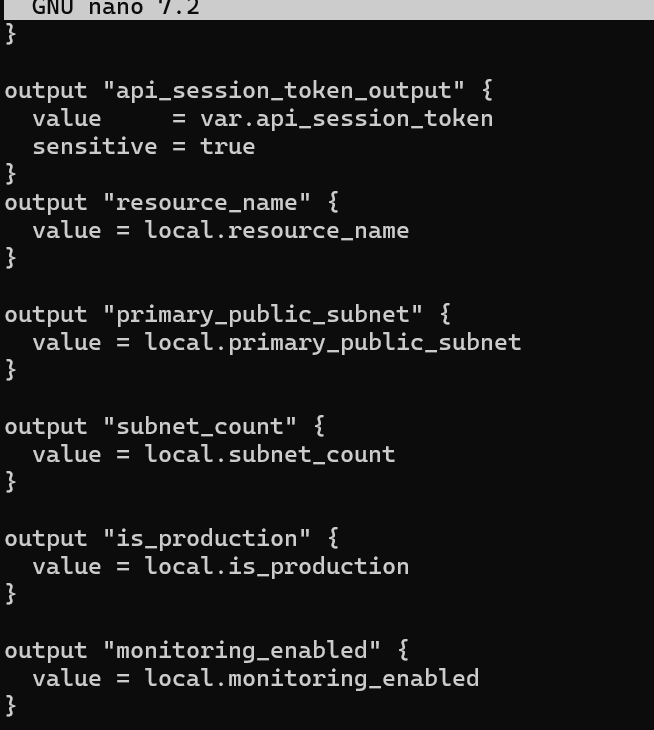


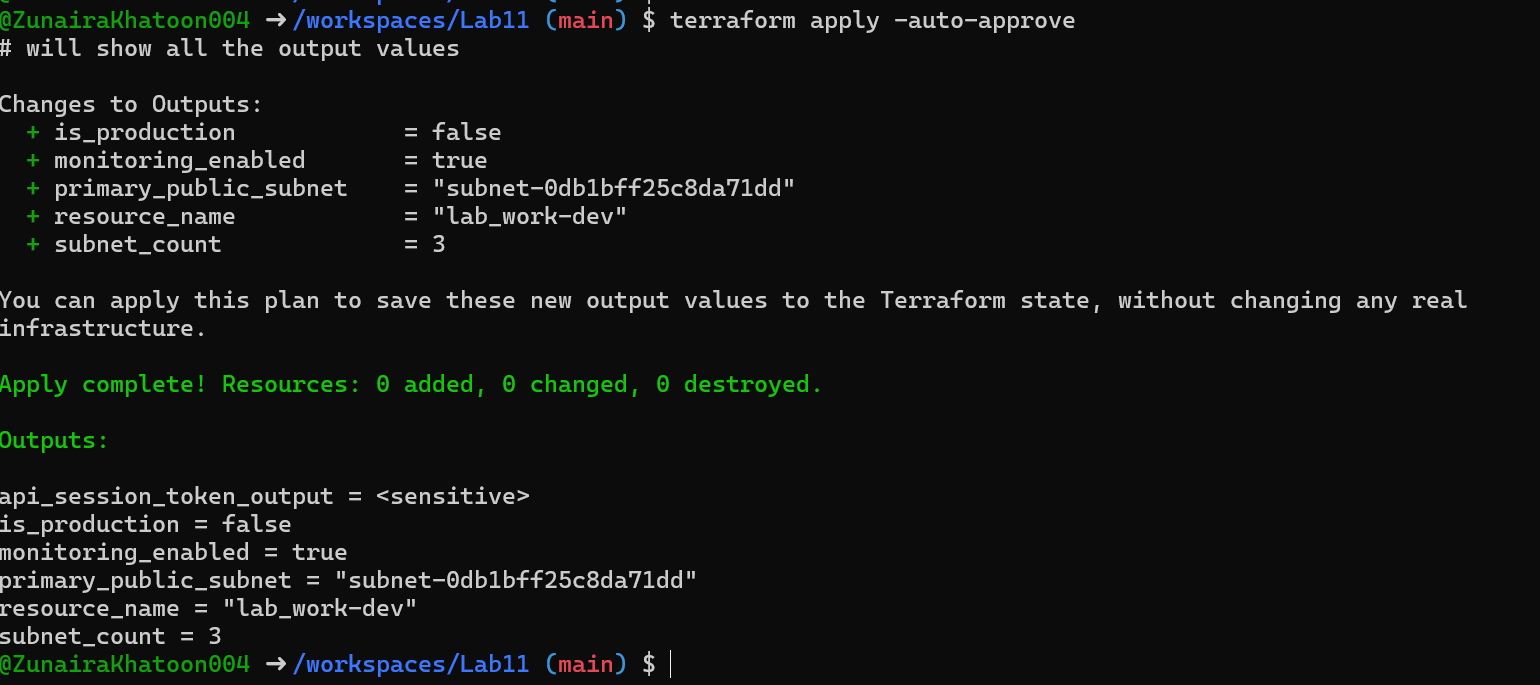


1. Create locals.tf with:



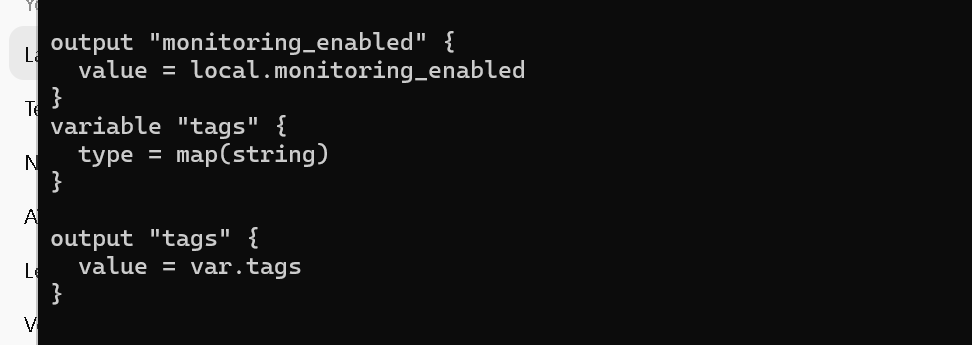
1. Add outputs to main.tf:



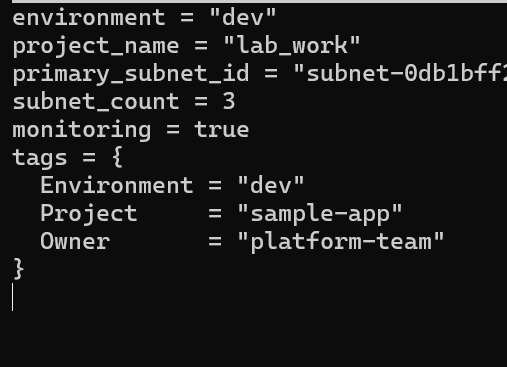


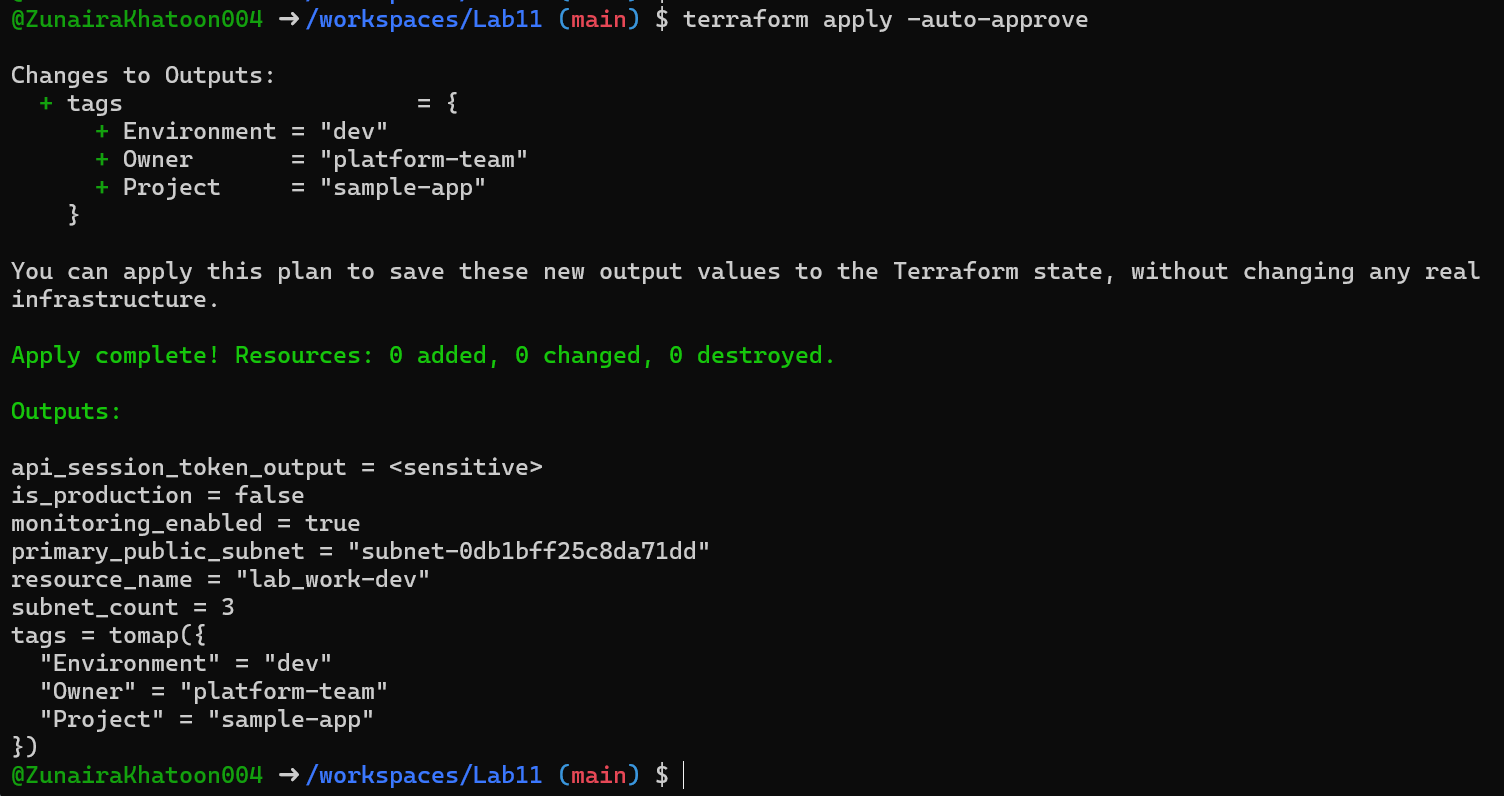
# [**Task 4 — Maps and Objects**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-4--maps-and-objects)

1. Map variable in main.tf:

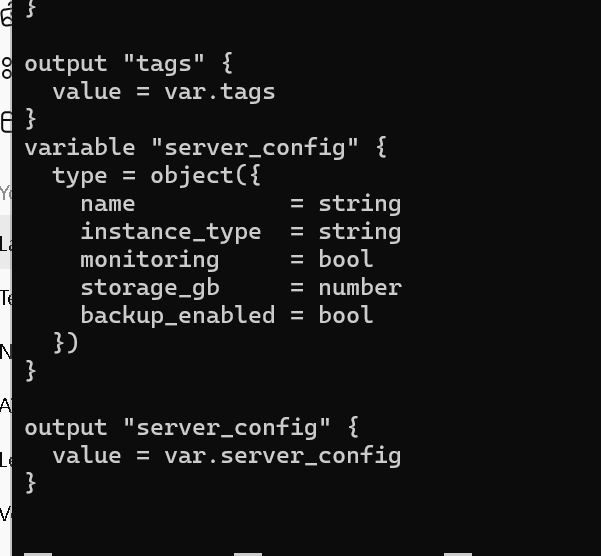


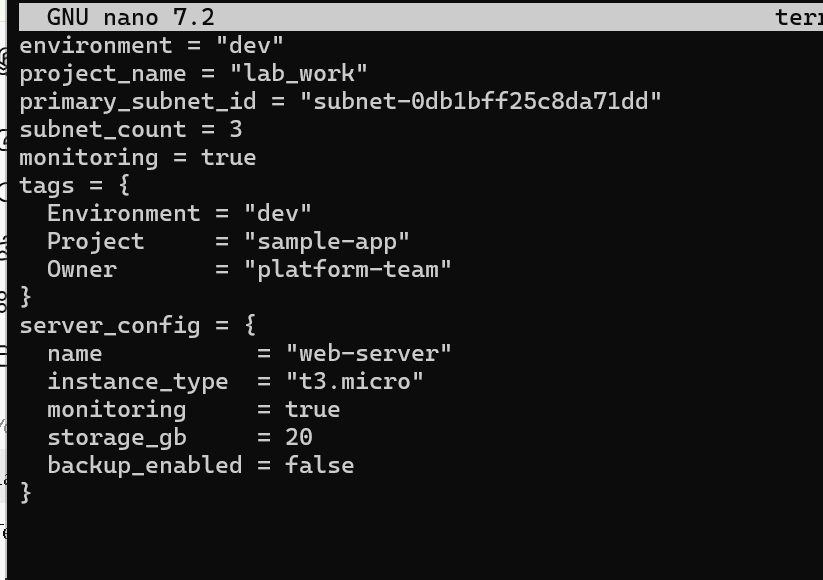
1. In terraform.tfvars:

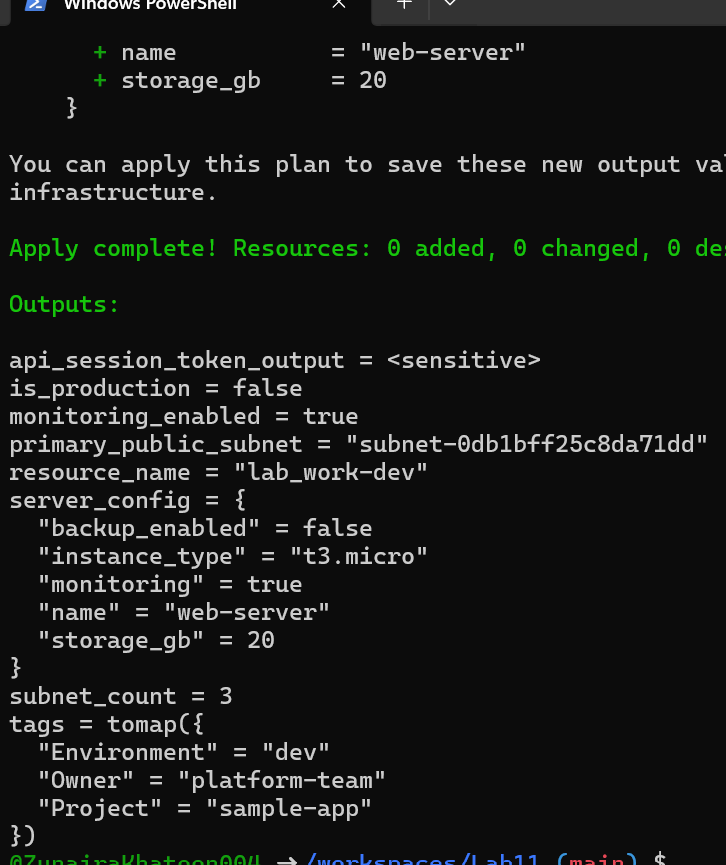




1. Define object variable:

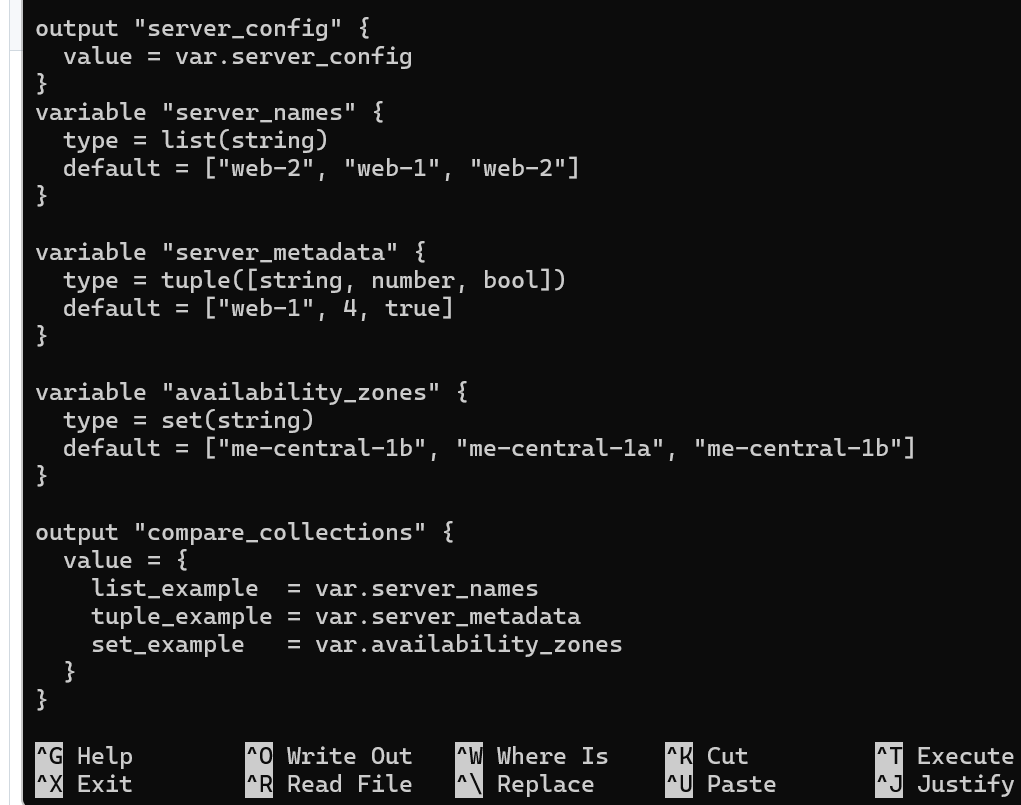




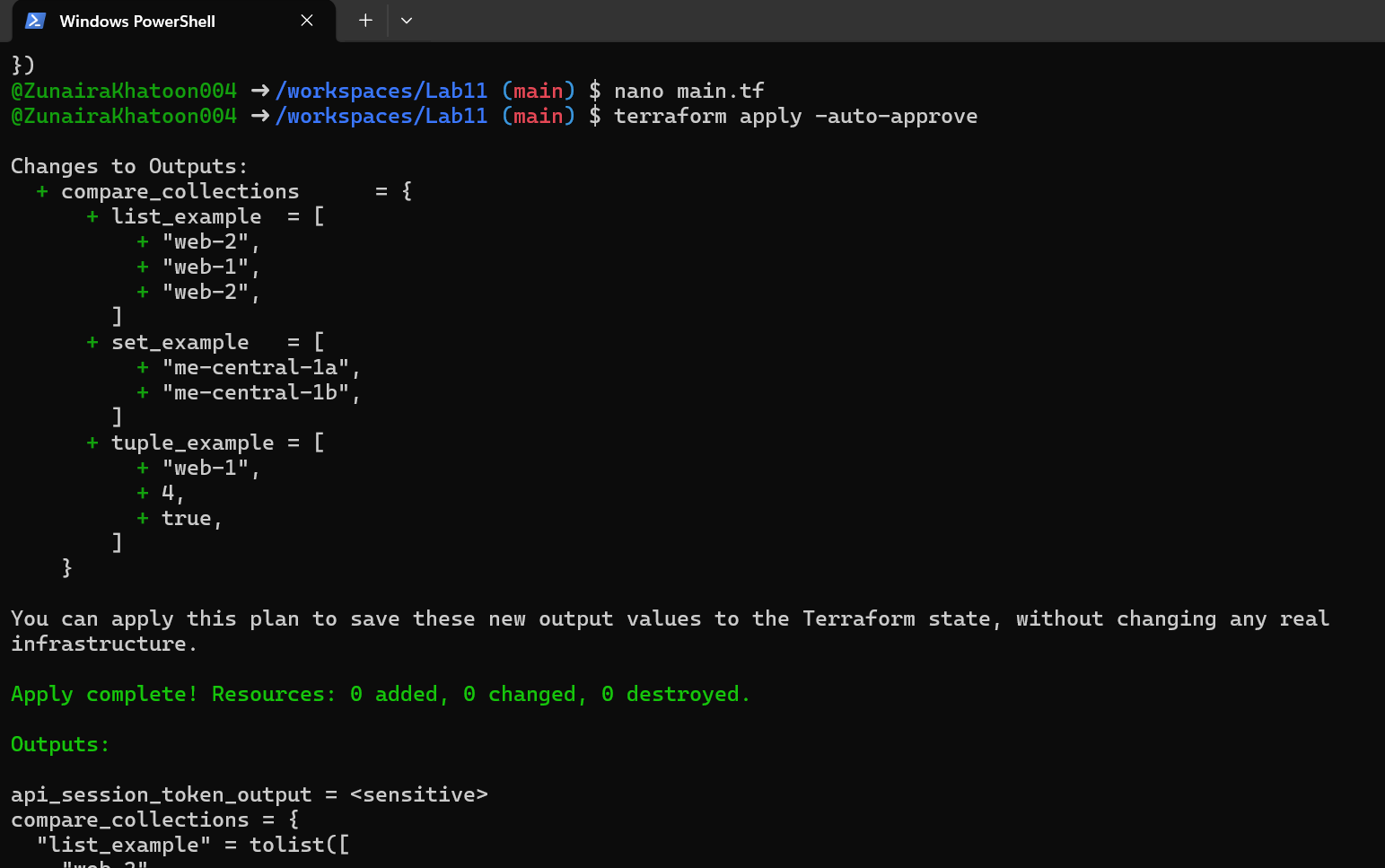


# [**Task 5 — Collections: list, tuple, set & mutation via locals**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-5--collections-list-tuple-set--mutation-via-locals)

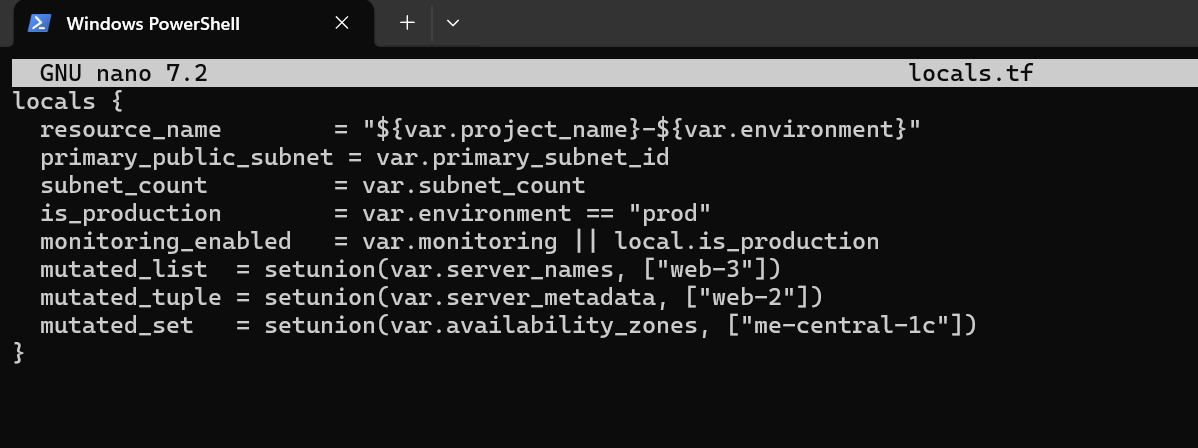
1. In main.tf define:



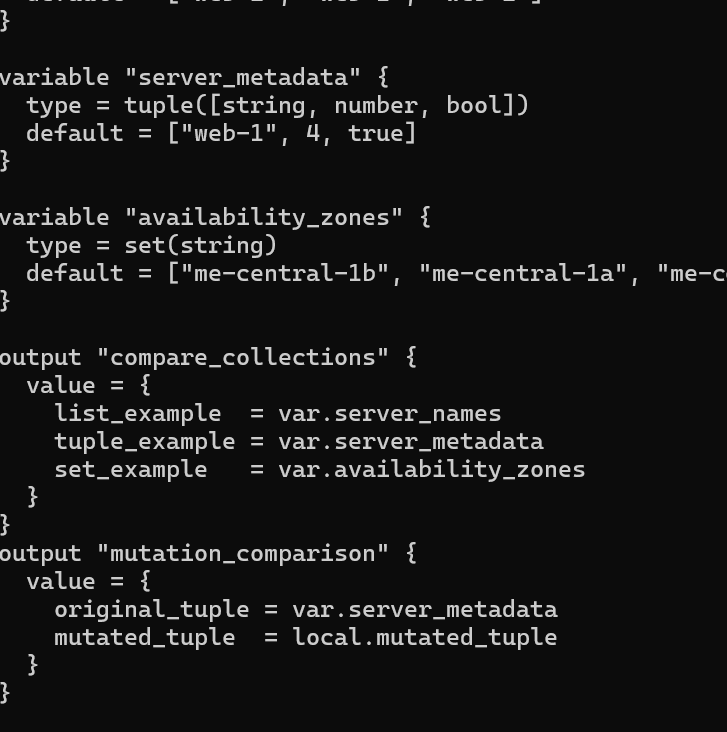
1. Run:

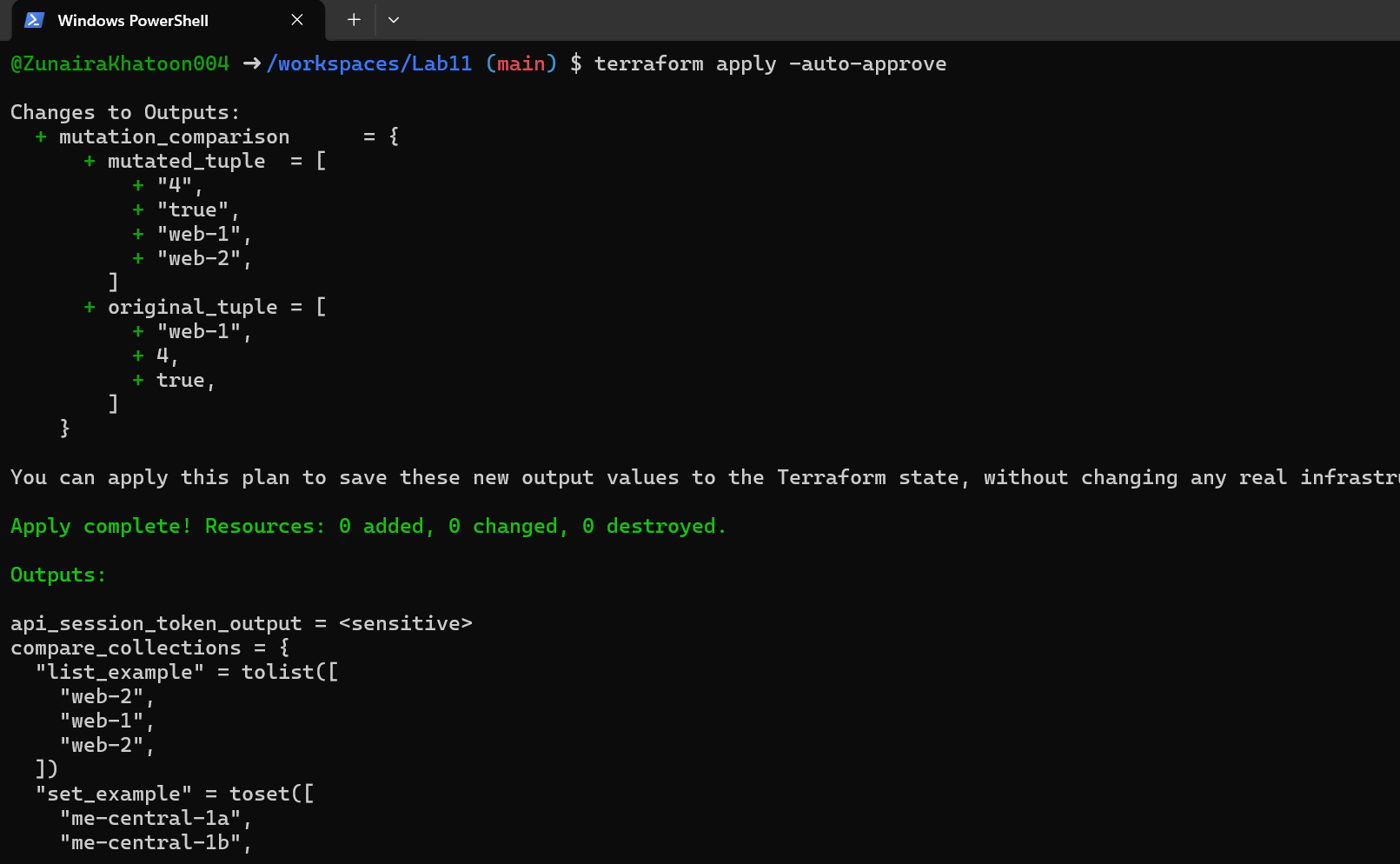


1. In locals.tf add mutations:



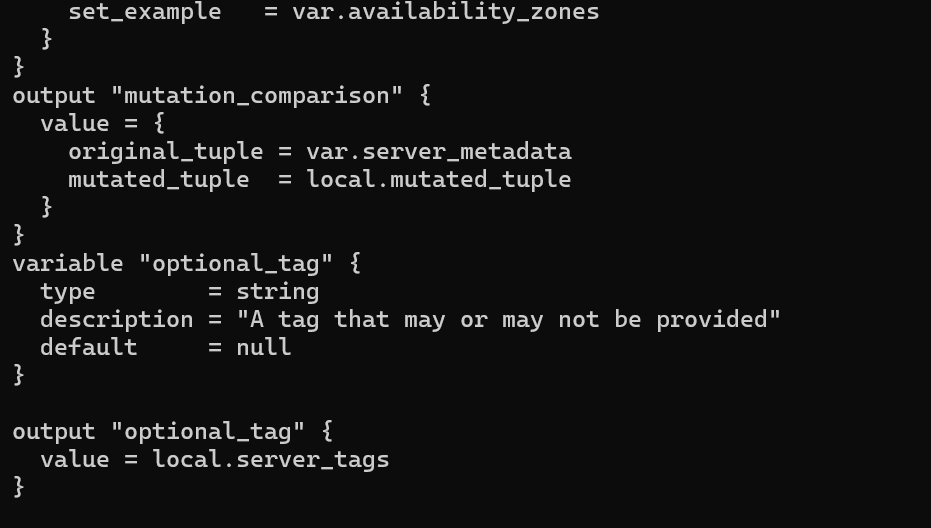
1. Add comparison output in main.tf:



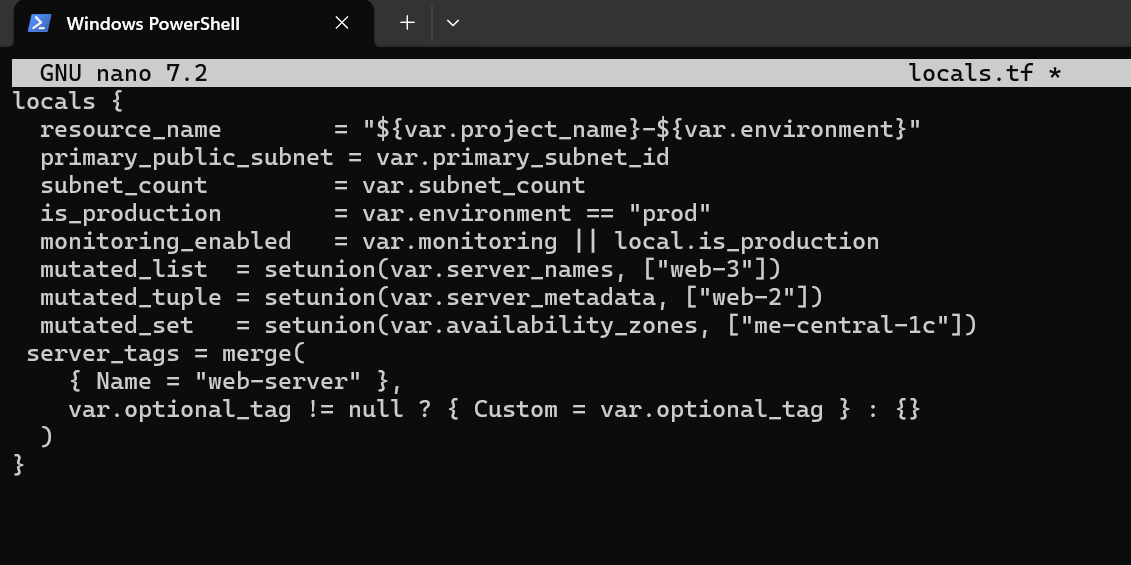


# [**Task 6 — Null, any type & dynamic values**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-6--null-any-type--dynamic-values)

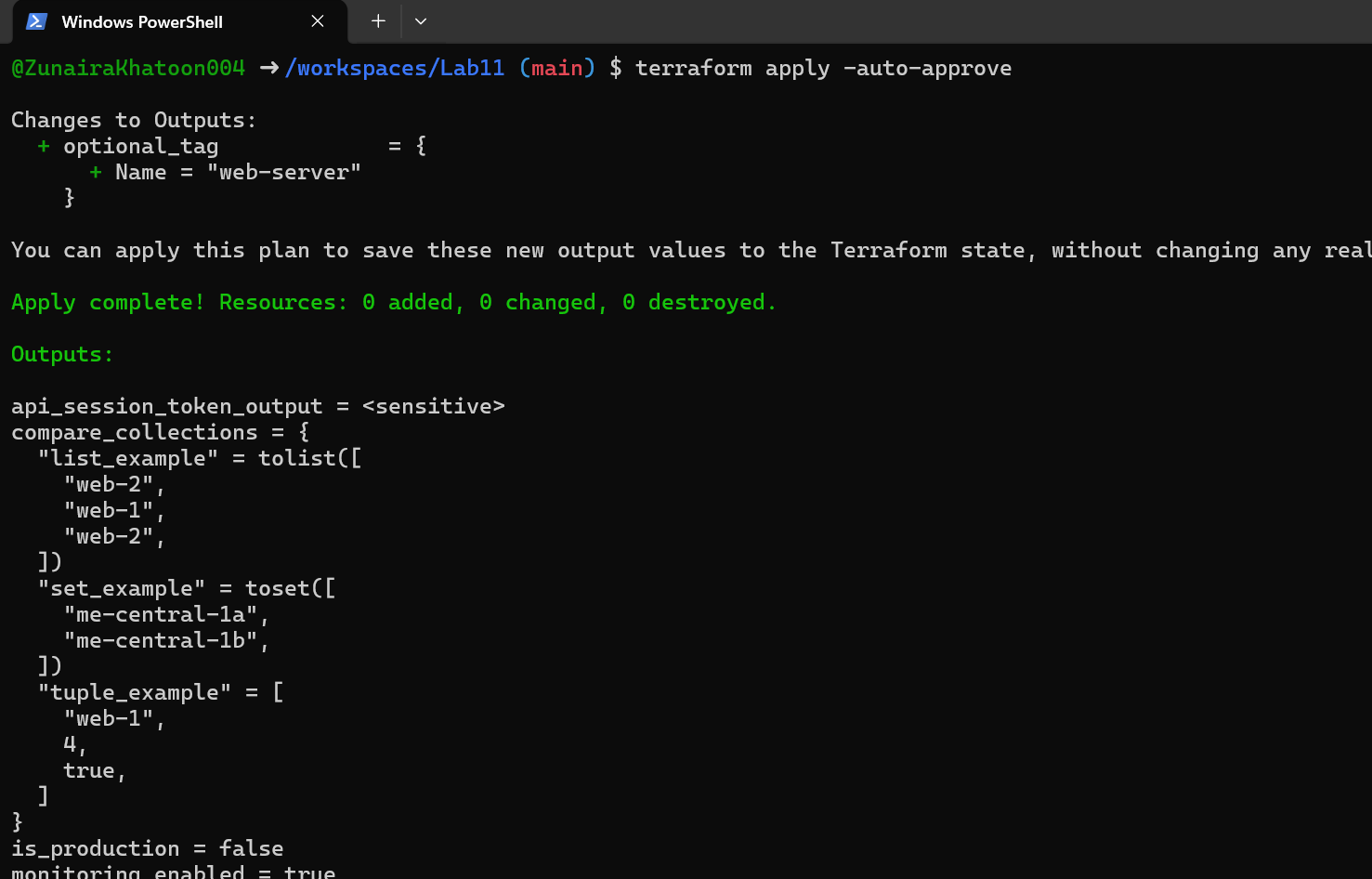
1. Null variable:



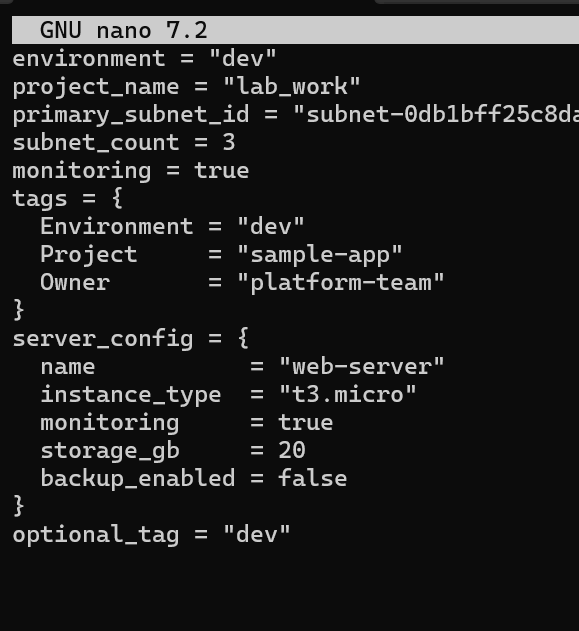
1. Merge tags in locals.tf:

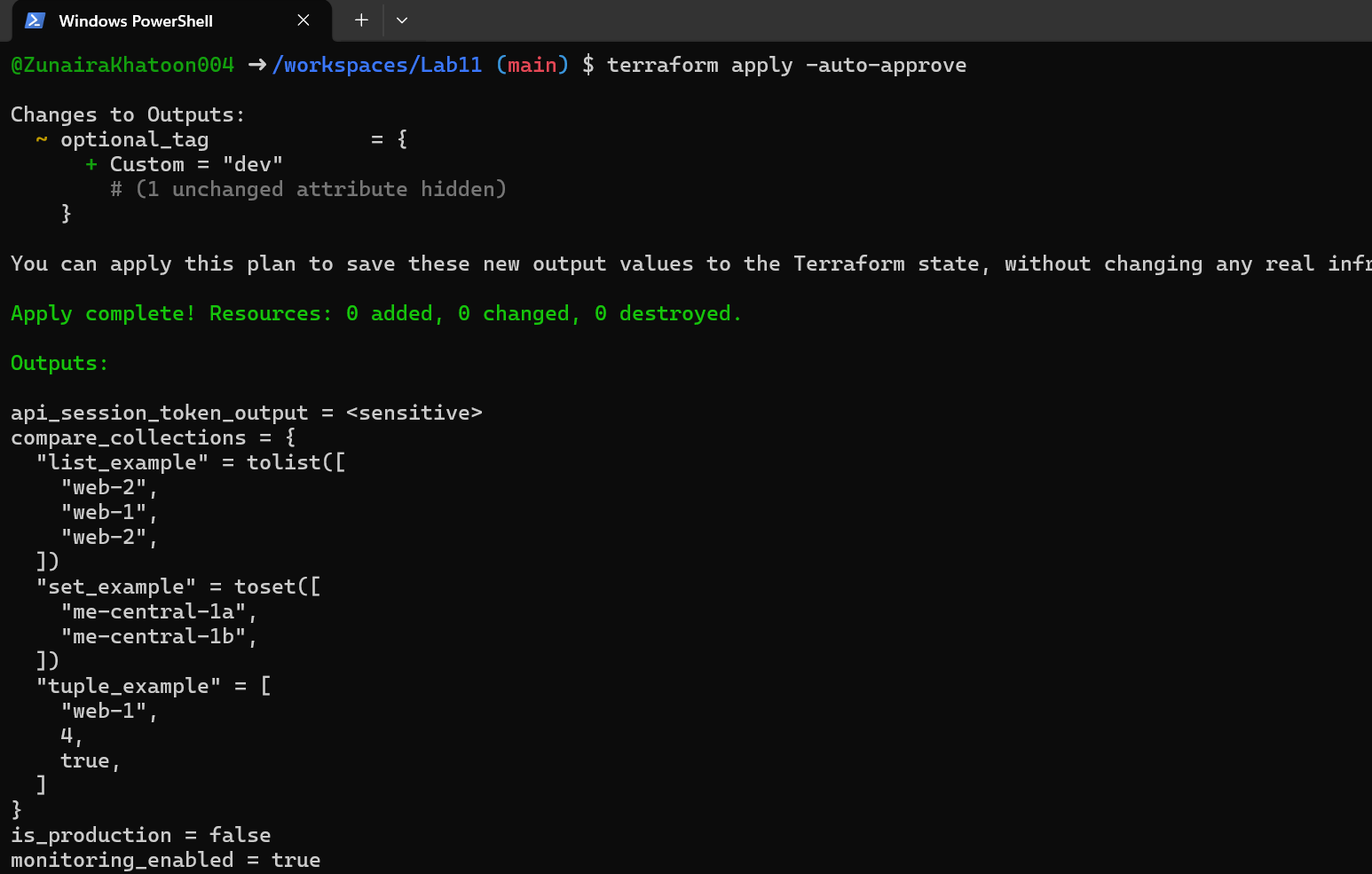


1. Output:

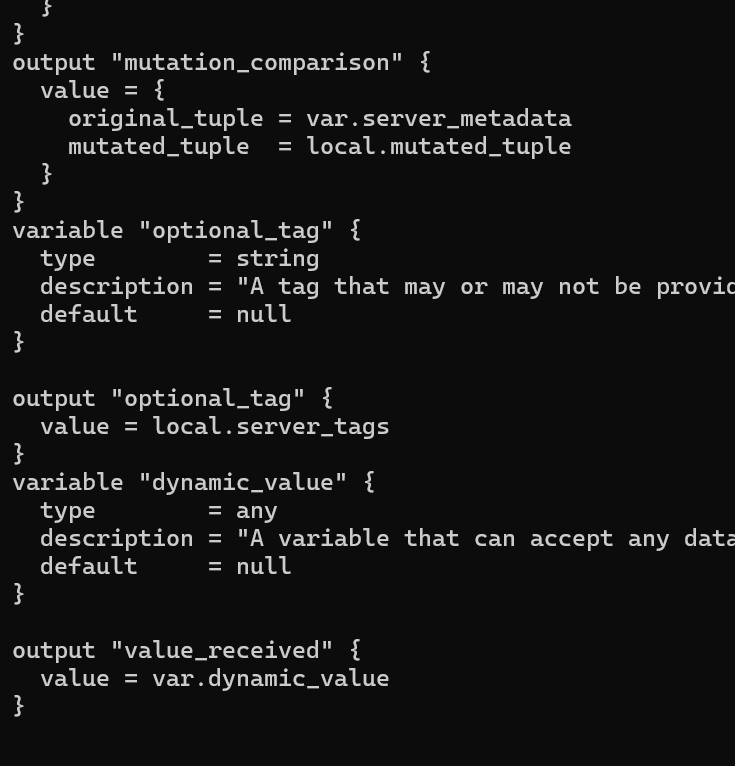


1. Add in terraform.tfvars:

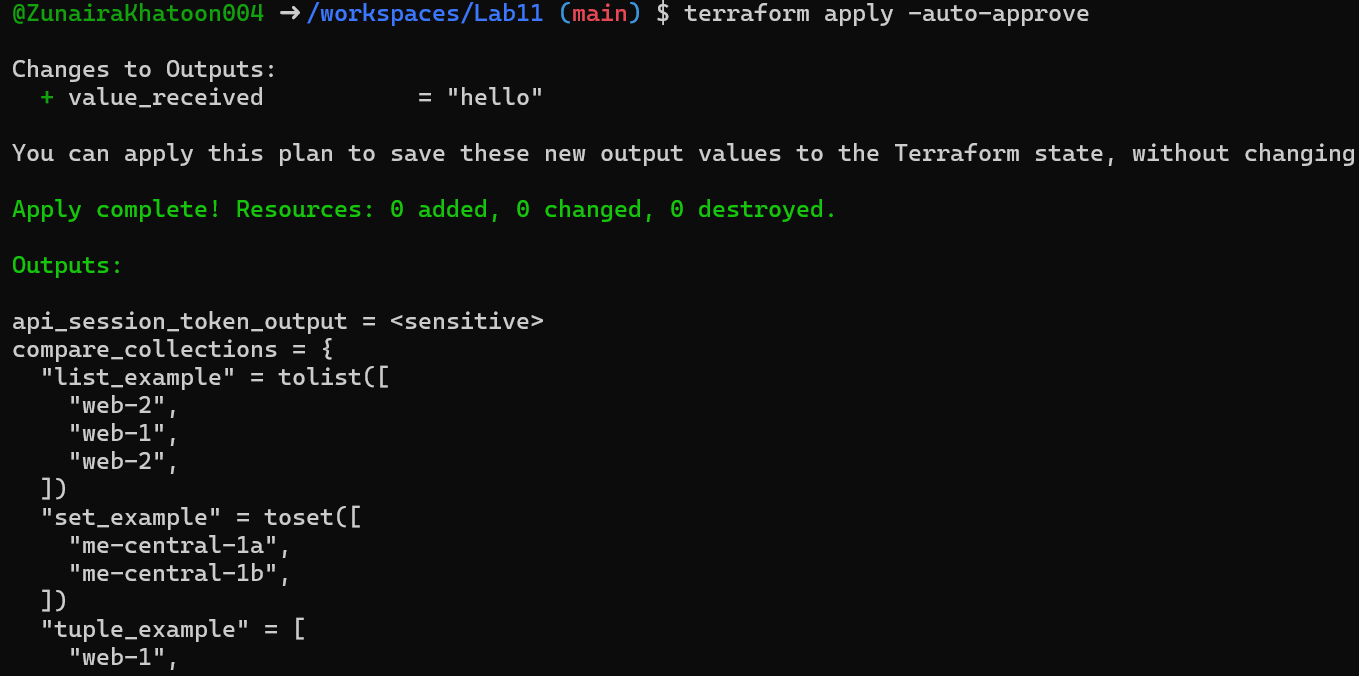


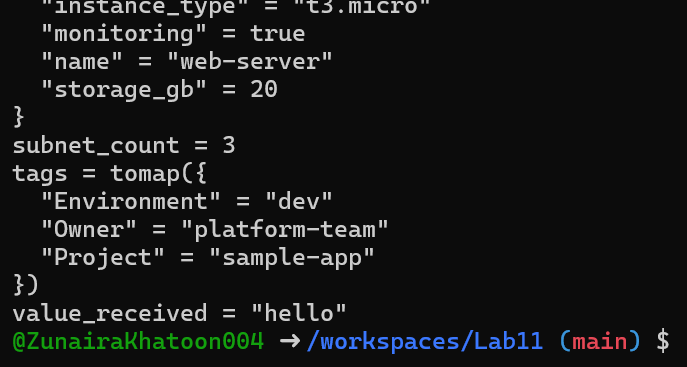


1. Any type variable:



1. Now test the dynamic (any) variable with different types. For each change, update terraform.tfvars with the specified value and run terraform apply -auto-approve, then capture the output.



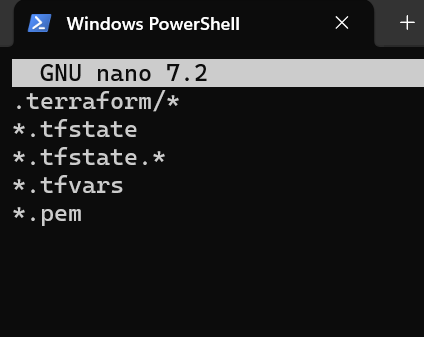


# [**Task 7 — Git ignore**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-7--git-ignore)

1. Create .gitignore:

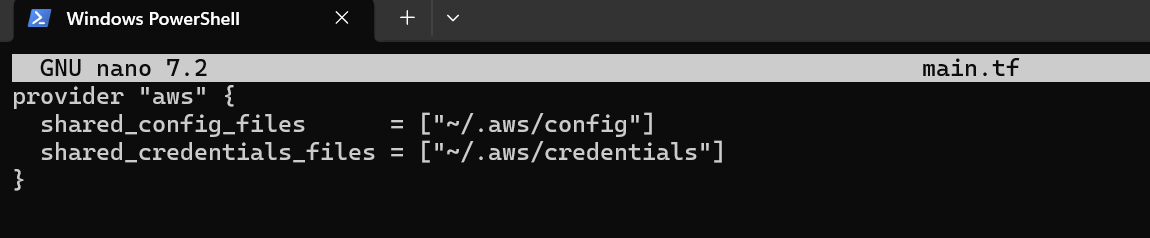


1. Add entries:

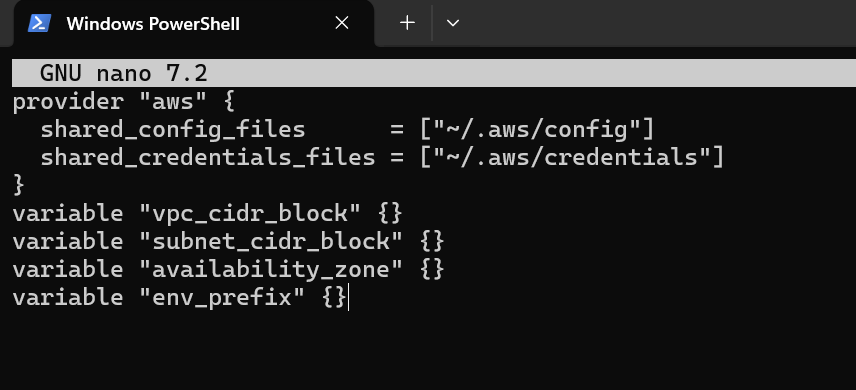


# [**Task 8 — Clean-up then build real infra (VPC, Subnet, IGW, routing, default route table)**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-8--clean-up-then-build-real-infra-vpc-subnet-igw-routing-default-route-table)

1. Clean previous files



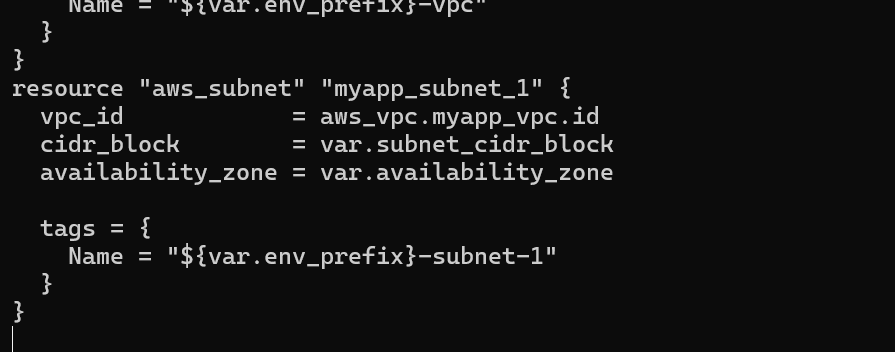
1. Define variables in main.tf



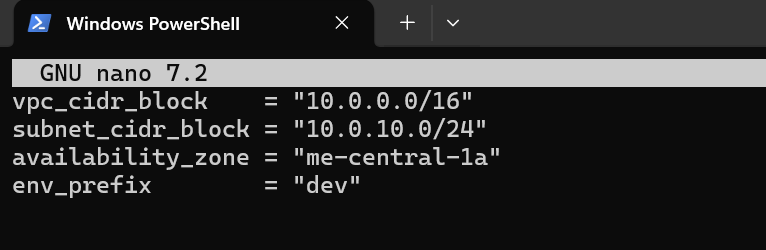
1. Create VPC in main.tf Add the VPC resource to main.tf:



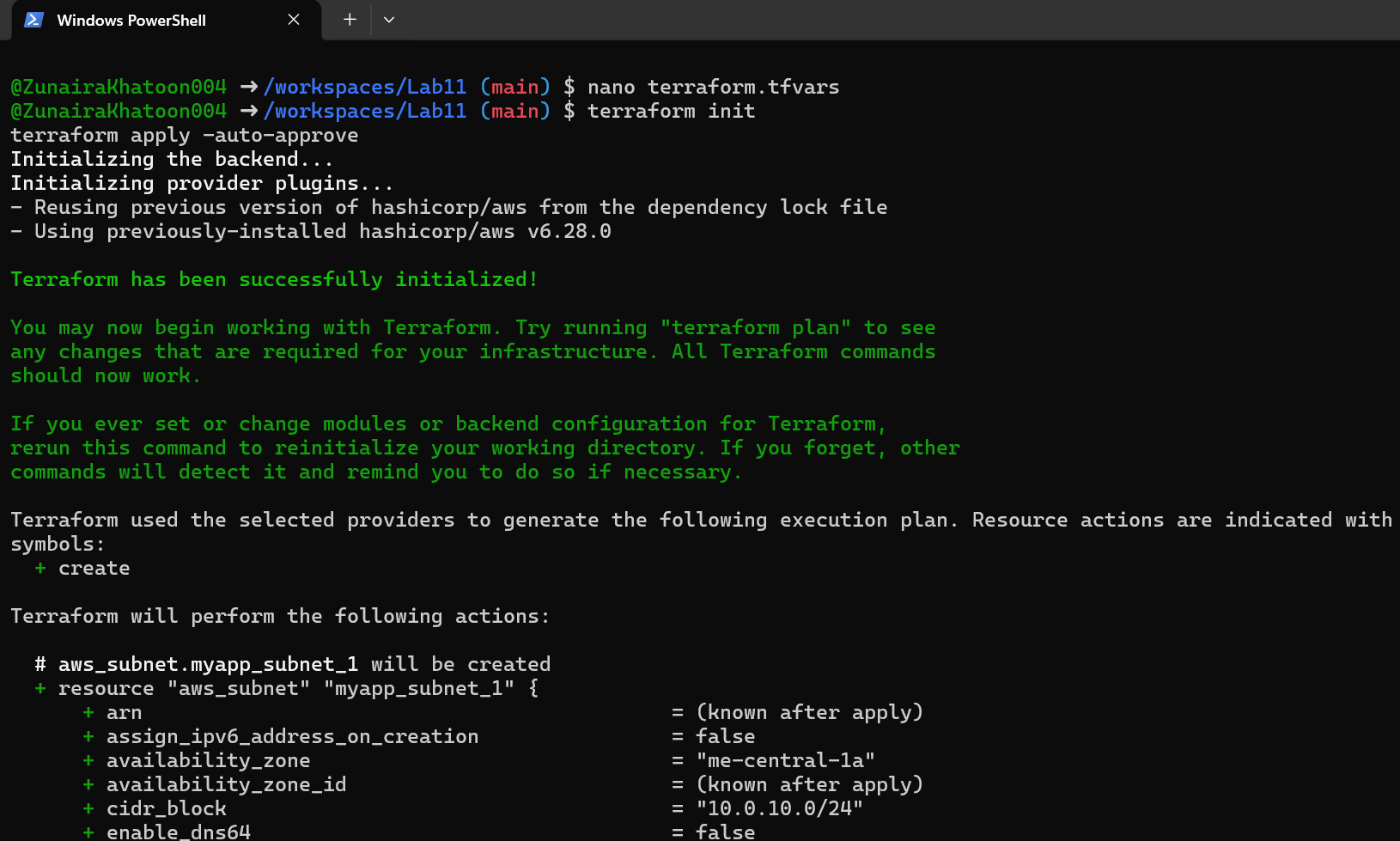
1. Create Subnet in the VPC Add the subnet resource to main.tf:



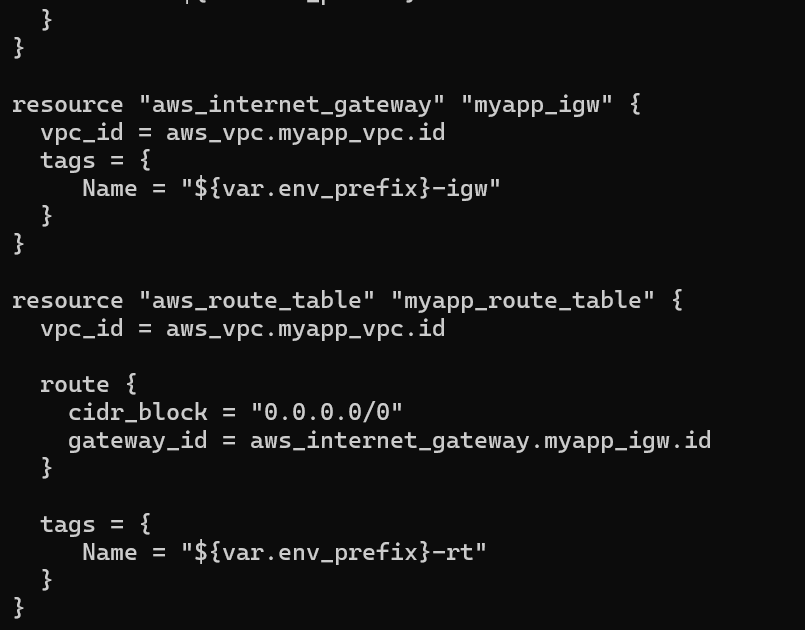
1. Populate terraform.tfvars In terraform.tfvars add:

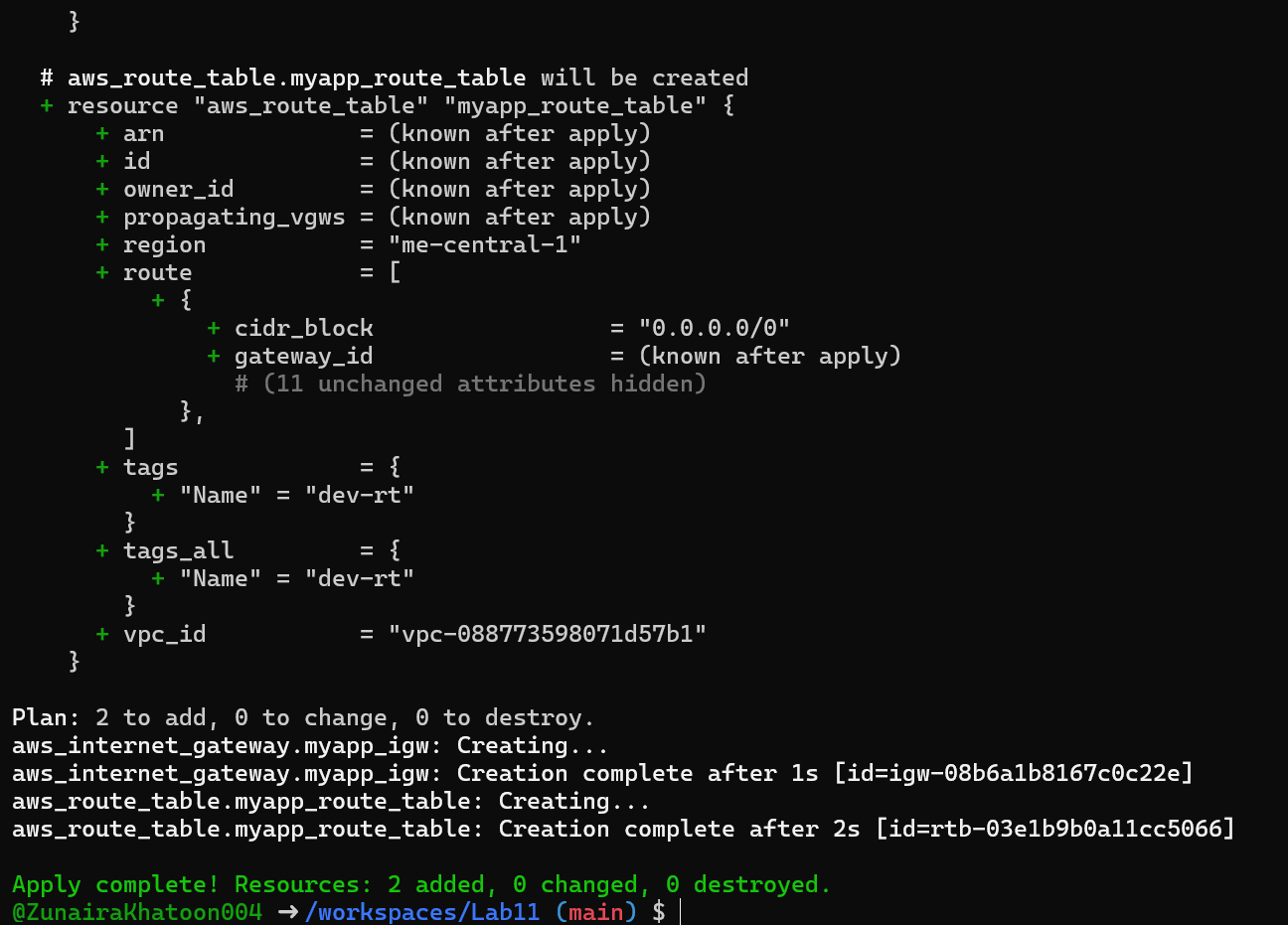


1. Apply to create VPC and Subnet Initialize (if needed) and apply:

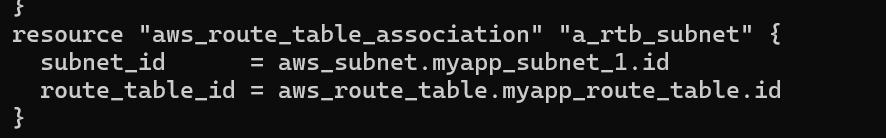


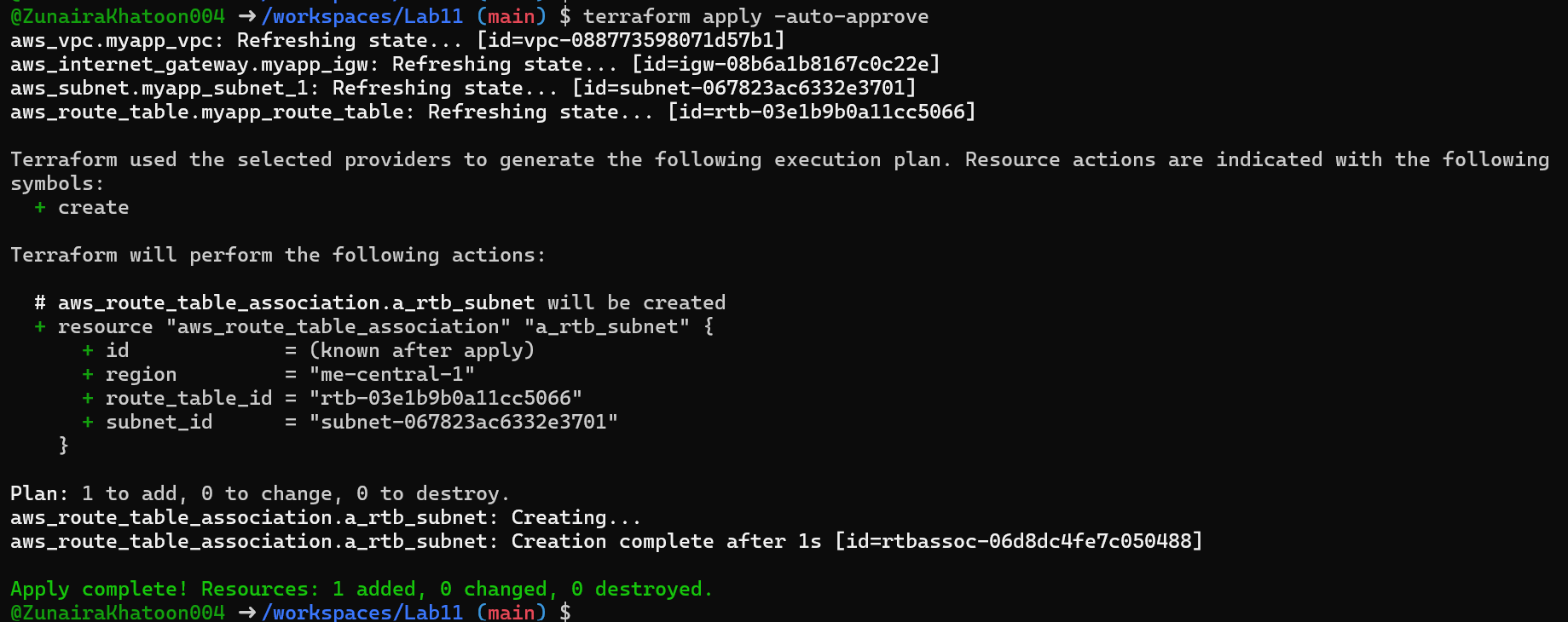
1. Create Internet Gateway and Route Table (custom) Add the Internet Gateway and a custom Route Table to main.tf:



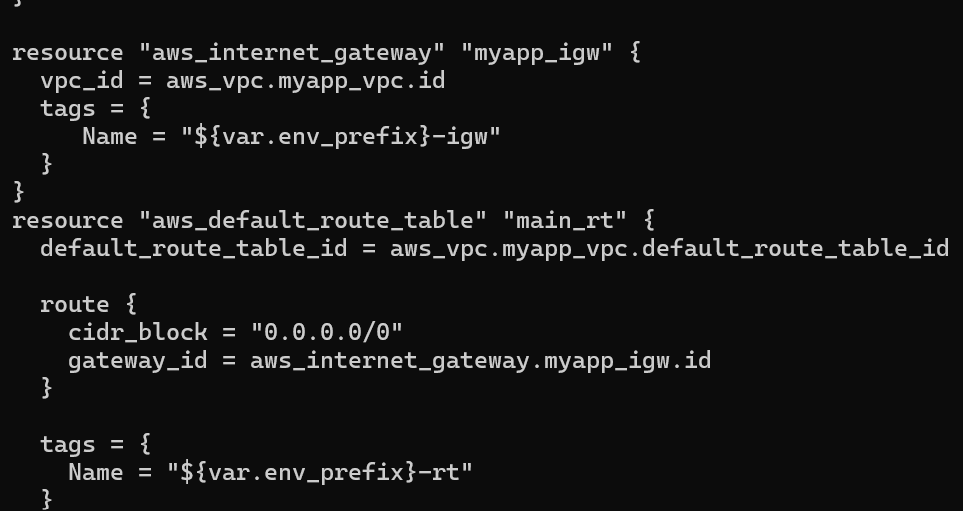


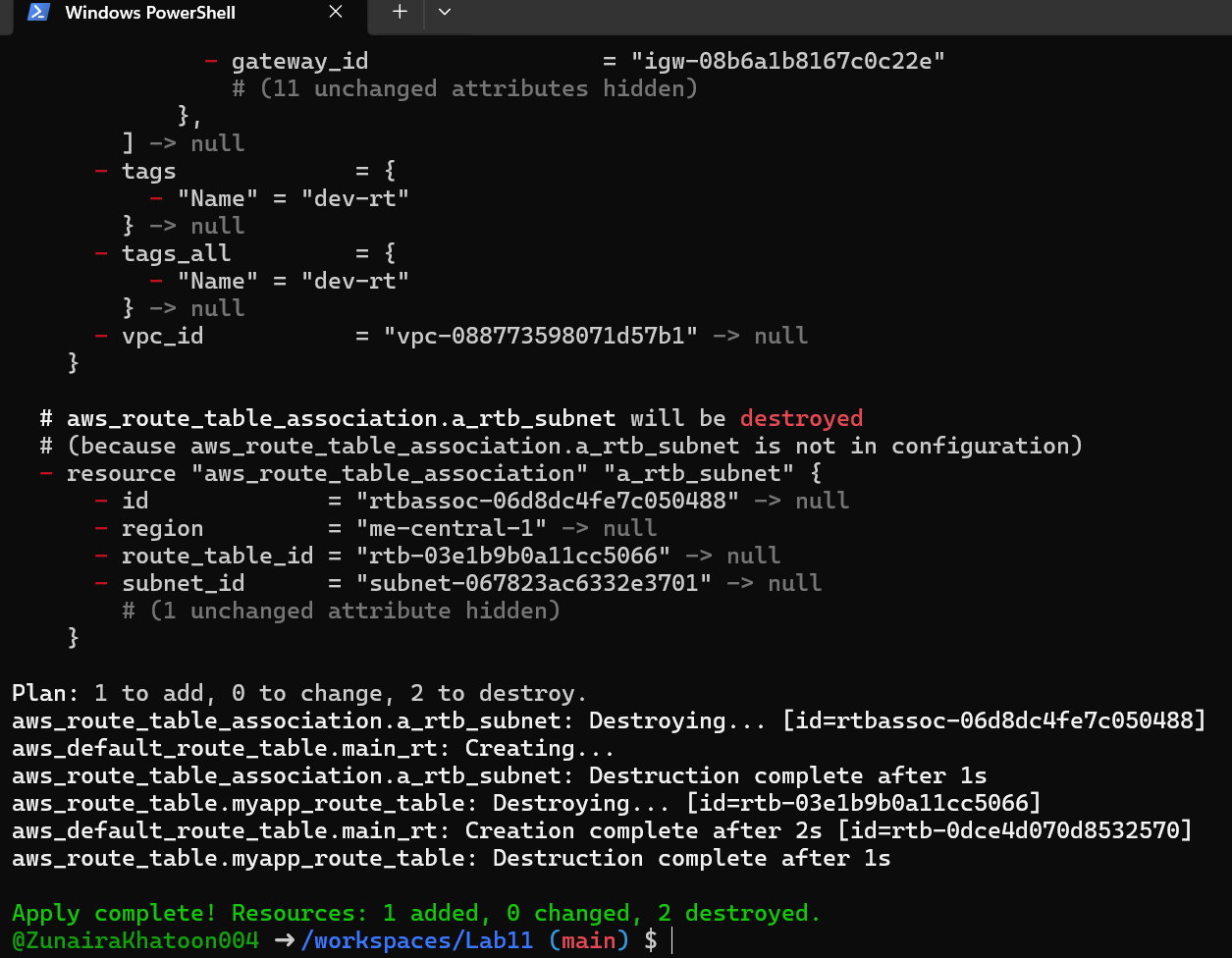
1. Associate the Route Table with the Subnet Add the association resource to main.tf:





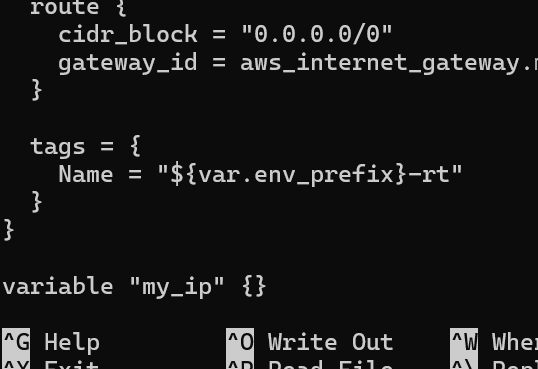
1. Switch to default route table (use VPC default route table) Now remove (or comment out) the custom route table and association resources from main.tf:



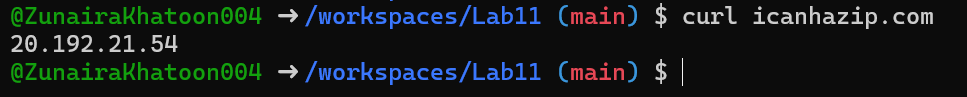


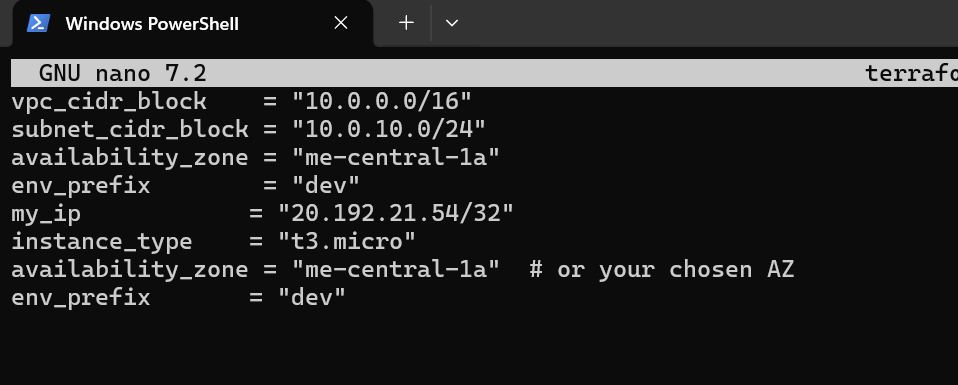
# [**Task 9 — Security Group, key pair, EC2 instance, user\_data & nginx**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#task-9--security-group-key-pair-ec2-instance-user_data--nginx)

1. Add variables to main.tf Add these variables to your main.tf:

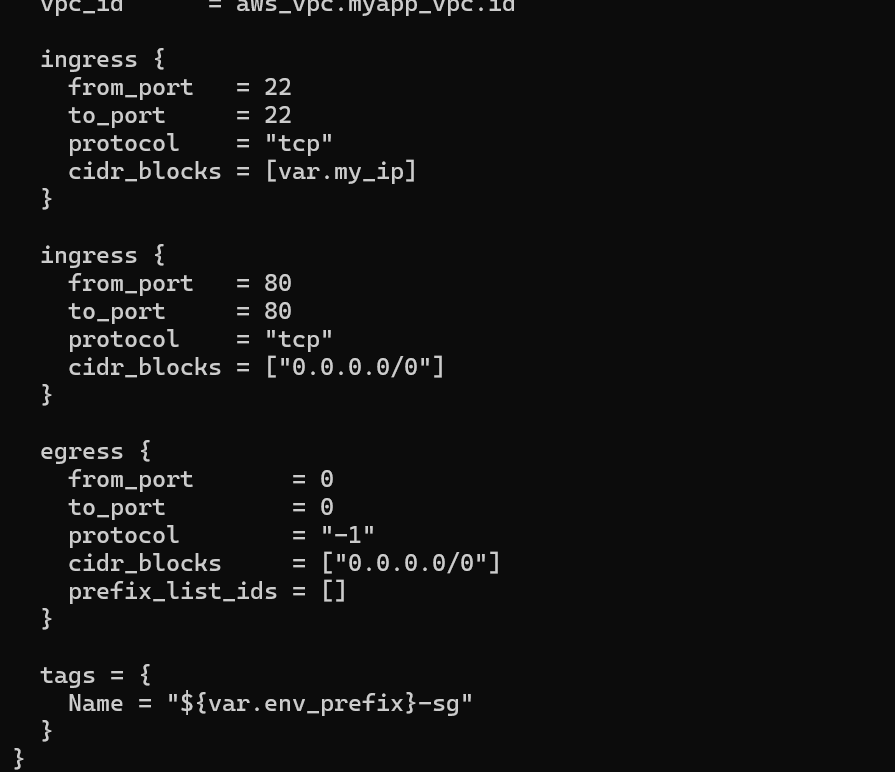


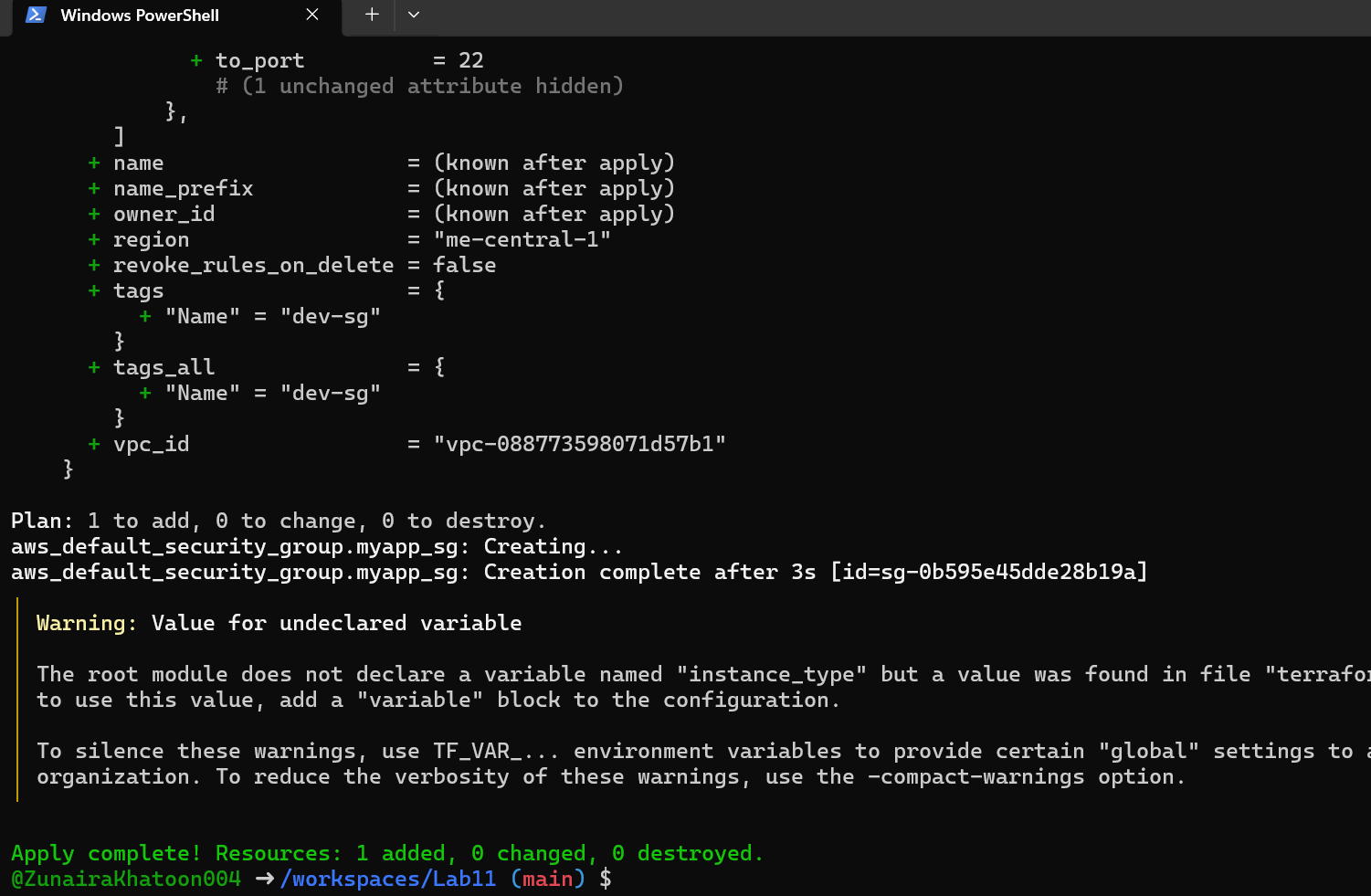
1. Get your public IP and set terraform.tfvars From the Codespace shell:



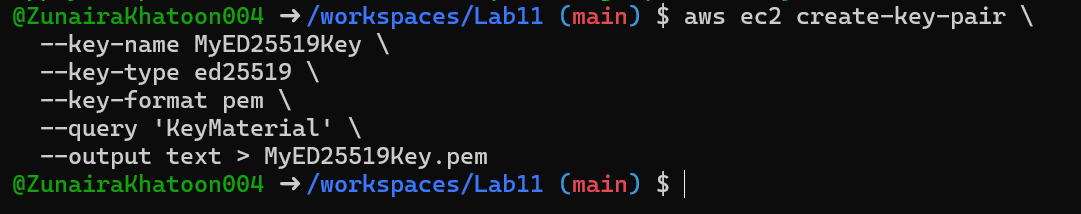


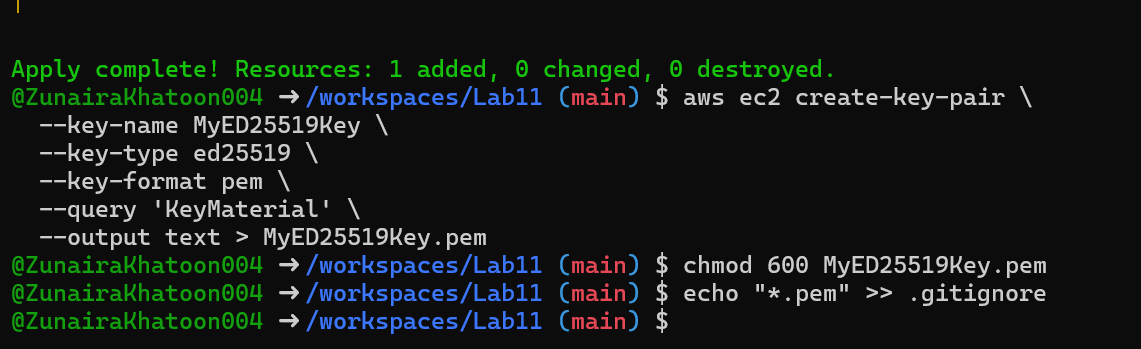
1. Create the Security Group (main.tf) Add this resource to main.tf:



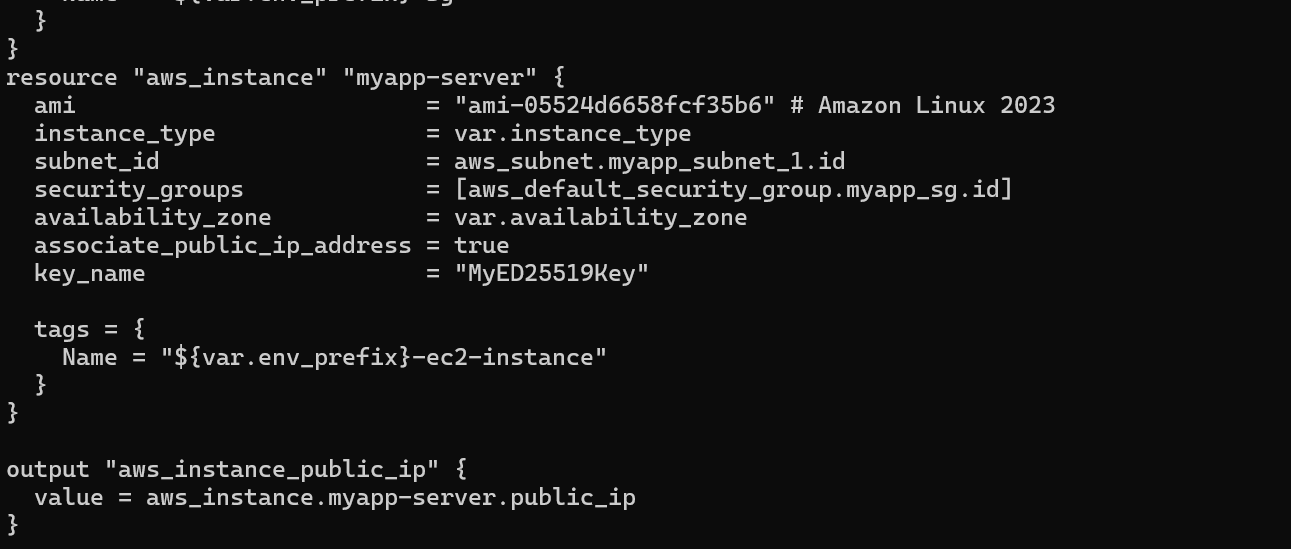


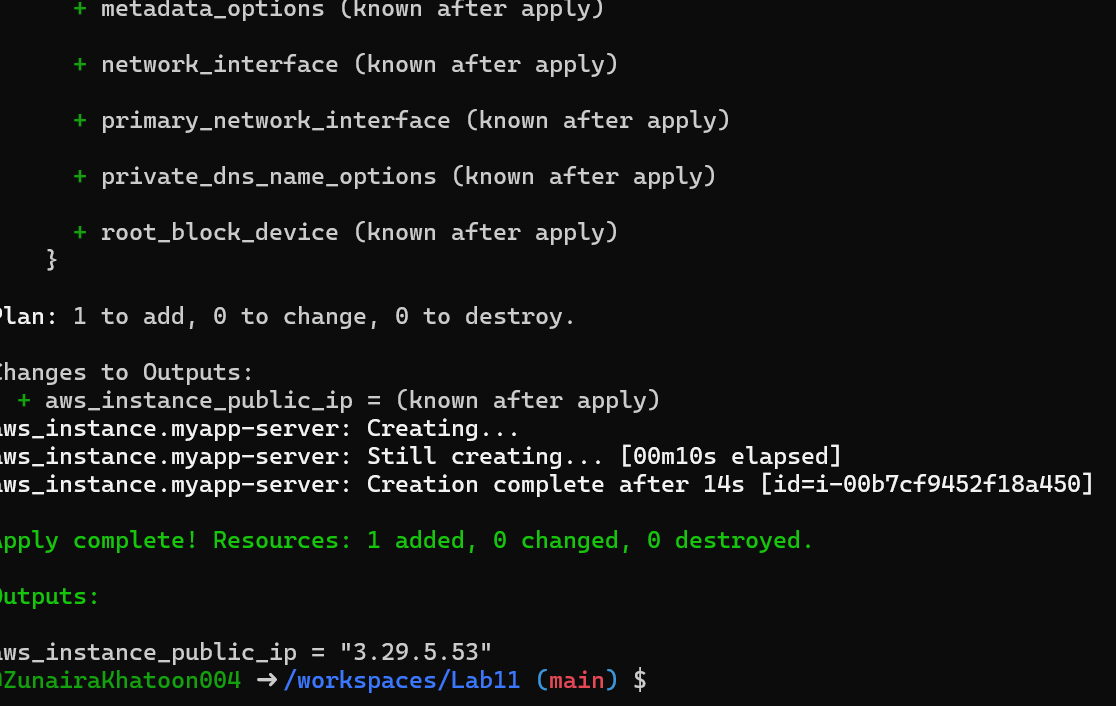
1. Create an AWS key pair and save locally Create a key pair and store the private key in your Codespace. Do NOT commit the .pem file.





1. Add EC2 instance resource (initial) Add the instance resource to main.tf (initially using the created key name):

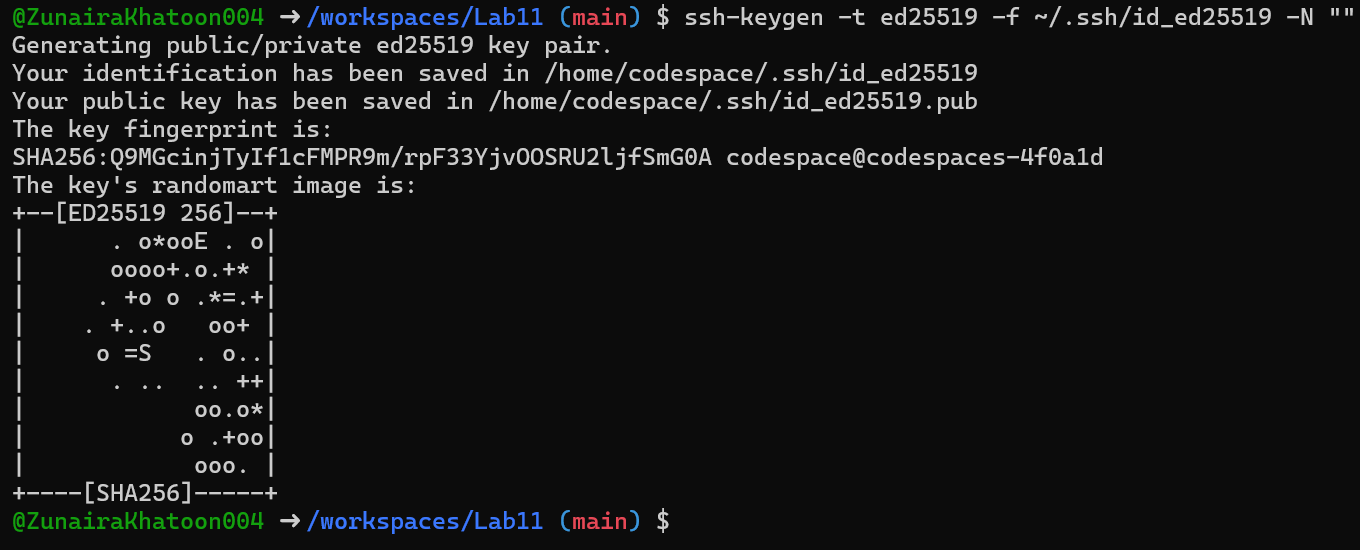




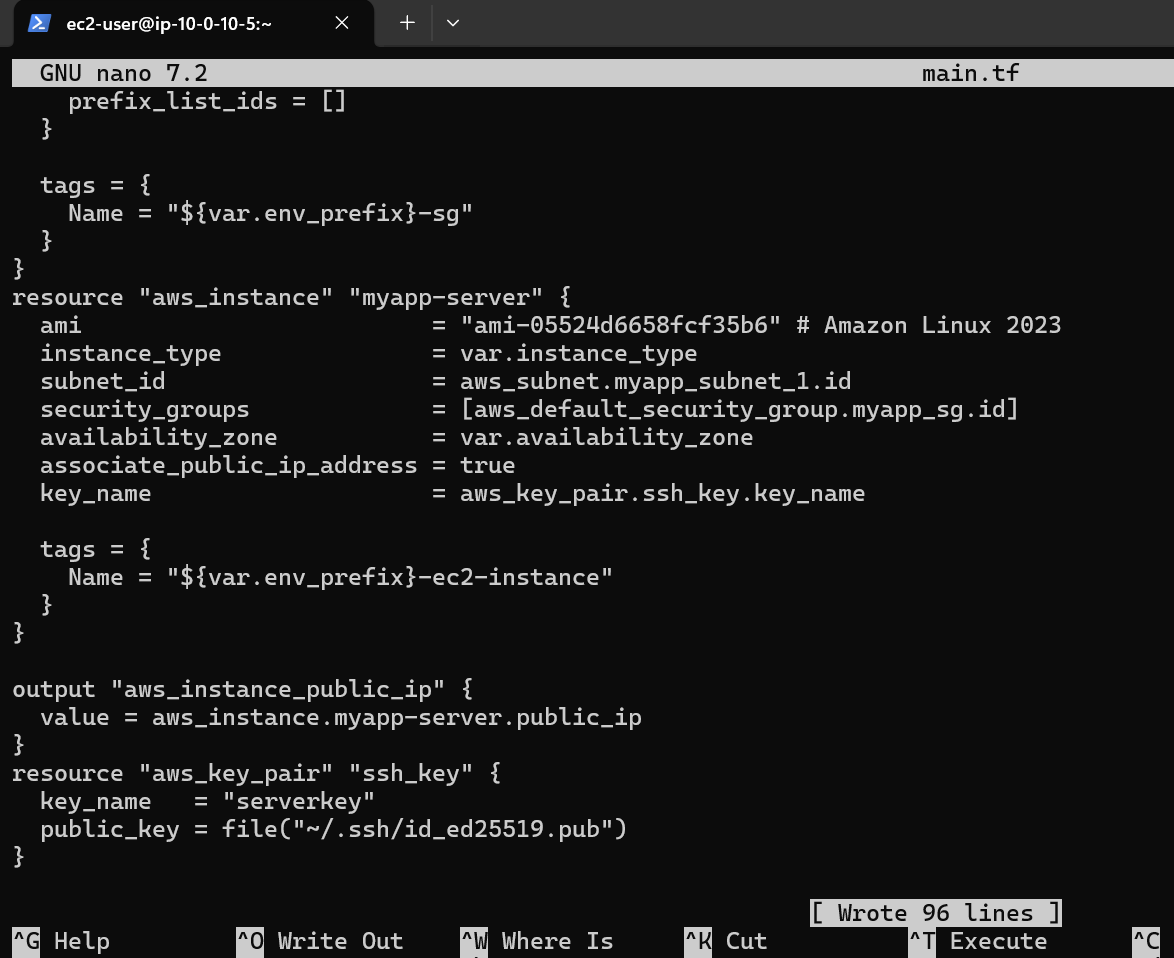
1. SSH into the instance (using MyED25519Key) From the Codespace:

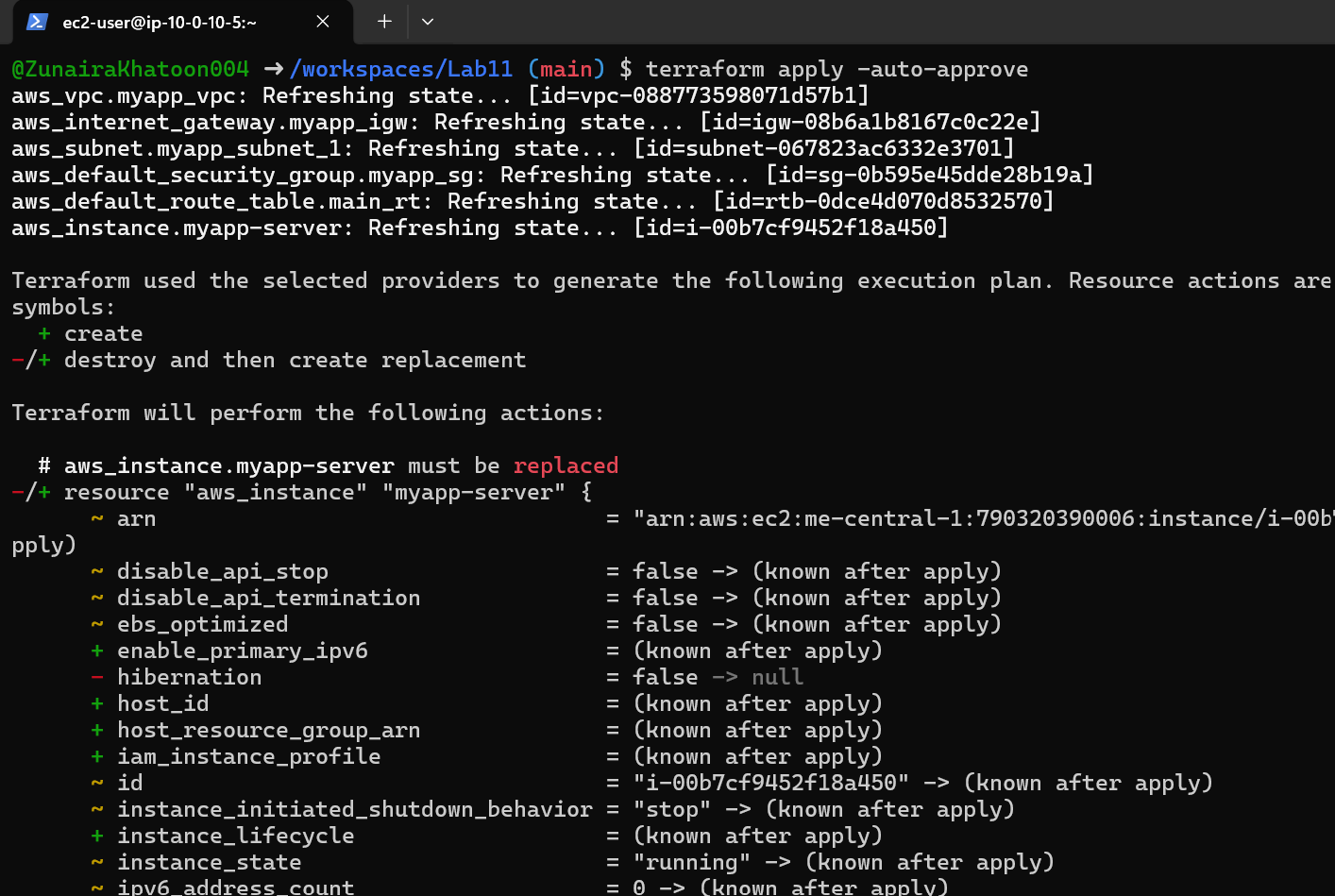


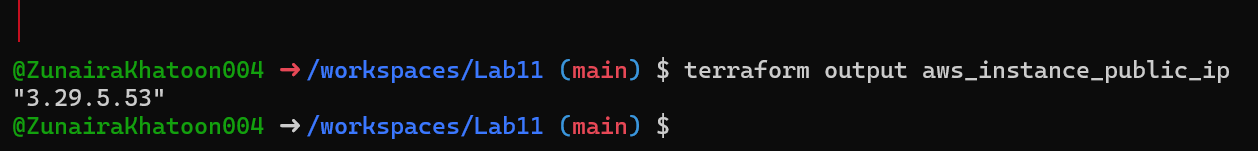
1. Generate a local SSH keypair and register it in AWS via Terraform On your Codespace, generate an SSH key pair (accept defaults or specify path):



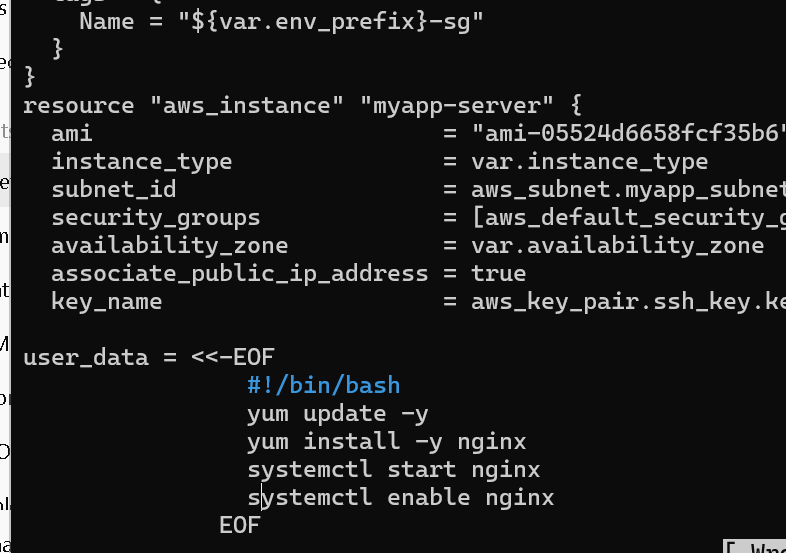
1. SSH using the newly registered key Now SSH with your generated private key (the default ssh client will pick up ~/.ssh/id\_ed25519):

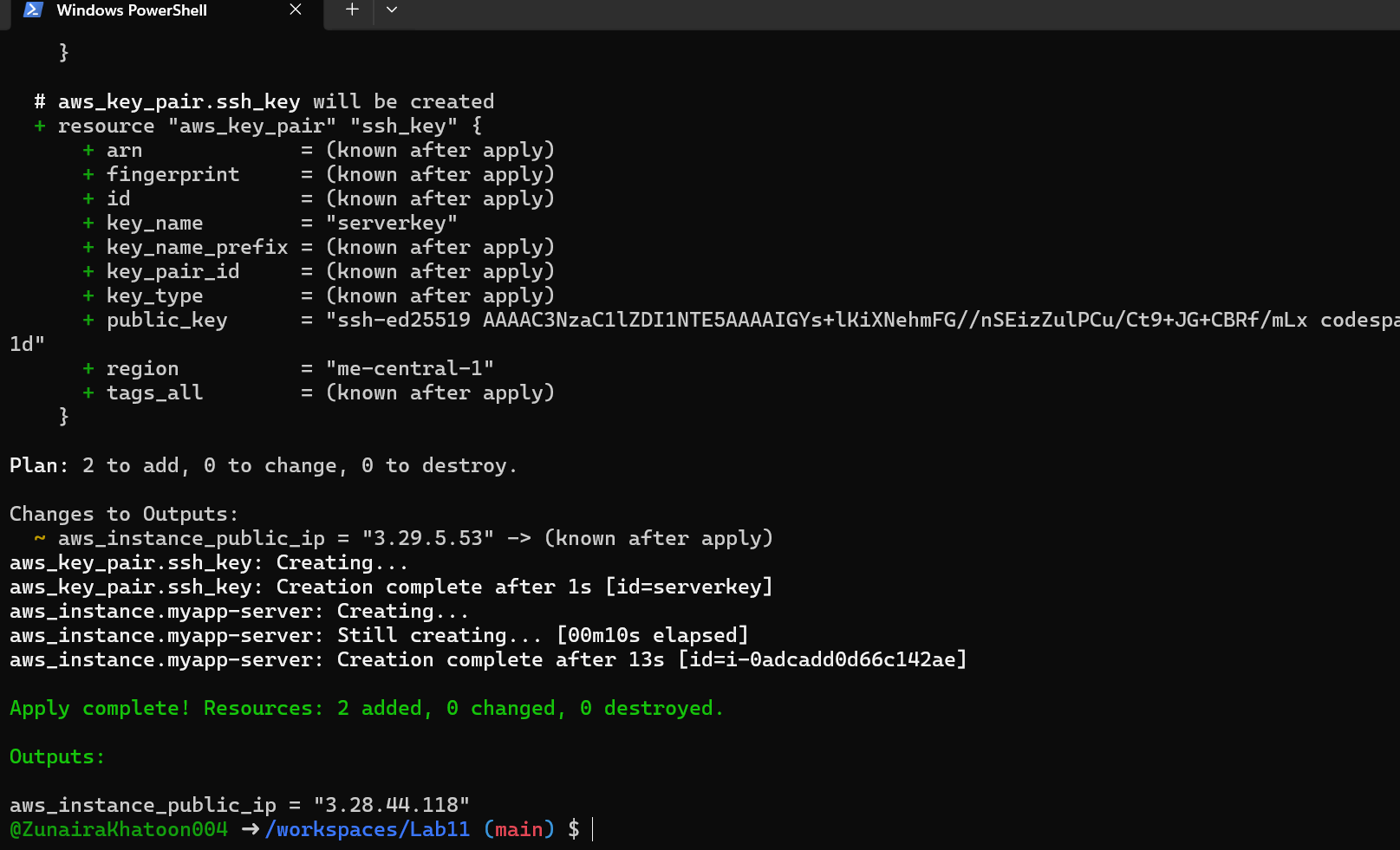


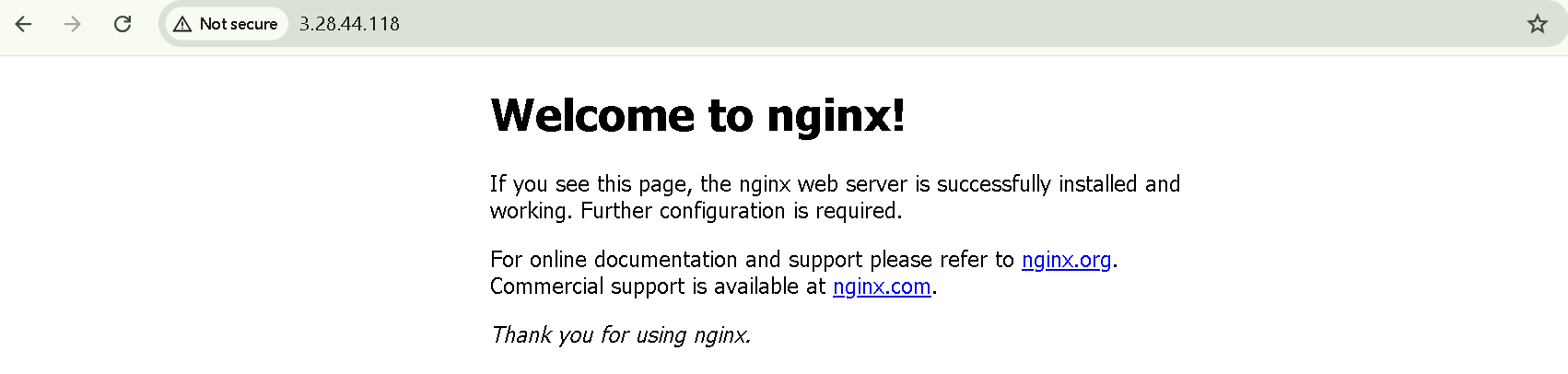




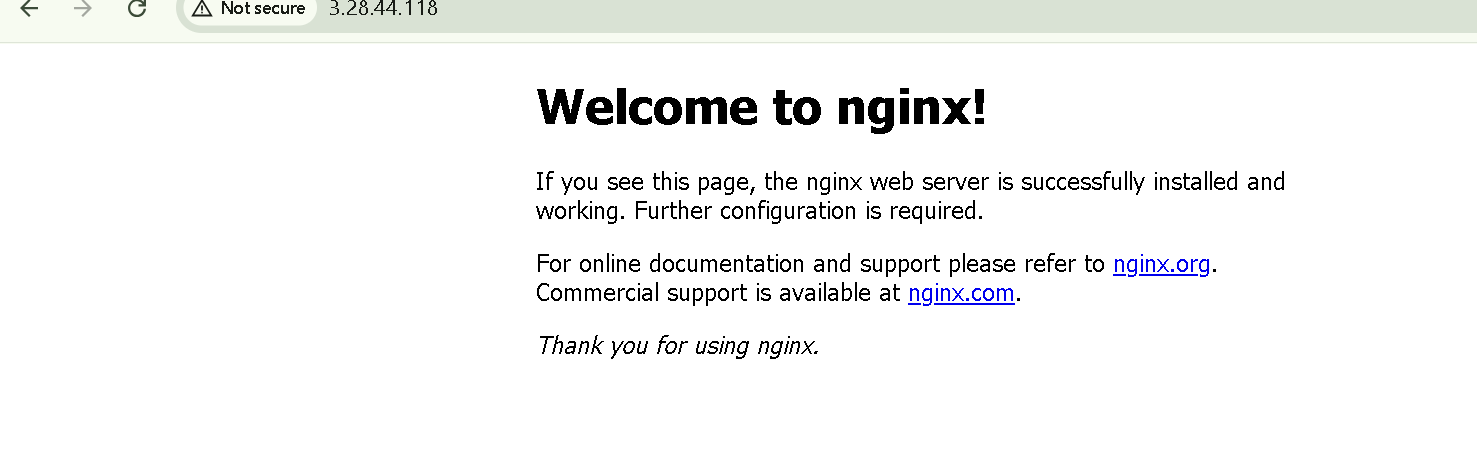
1. Install nginx via inline user\_data Modify the aws\_instance resource to include inline user\_data:





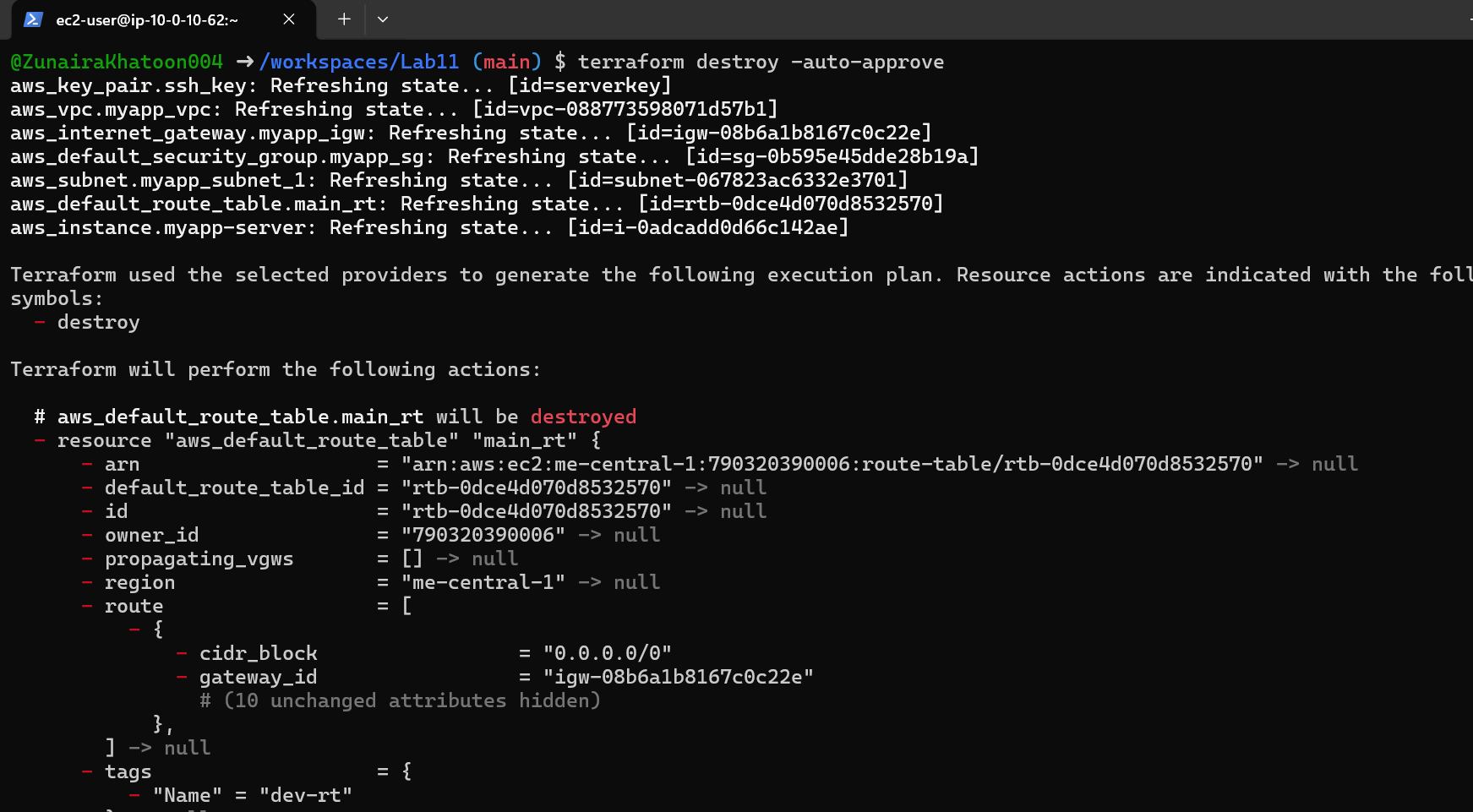


1. Use an external script for user\_data Create a script file in the Codespace:

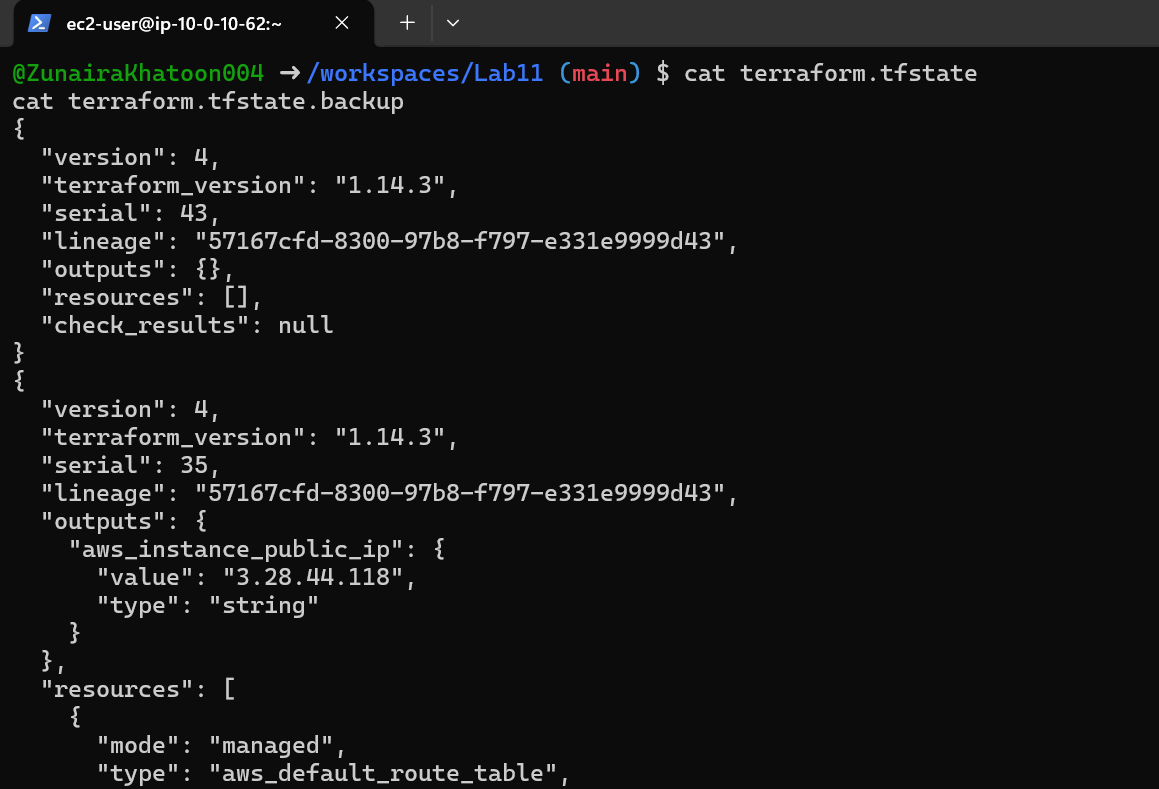


# [**Cleanup — Destroy resources & verify state**](https://github.com/WaqasSaleem97/CC_F25/tree/main/Lab_Tasks/Lab_11#cleanup)

1. Destroy all resources:



1. Verify state files:



1. Ensure no sensitive files are committed.

