**Lab Exercise 6– Terraform Multiple tfvars Files**

Objective:

Learn how to use multiple tfvars files in Terraform for different environments.

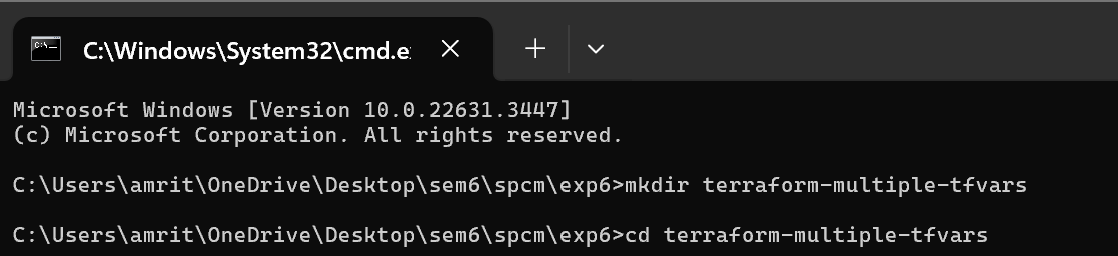
Prerequisites:

• Terraform installed on your machine.

• Basic knowledge of Terraform configuration and variables.

Steps:

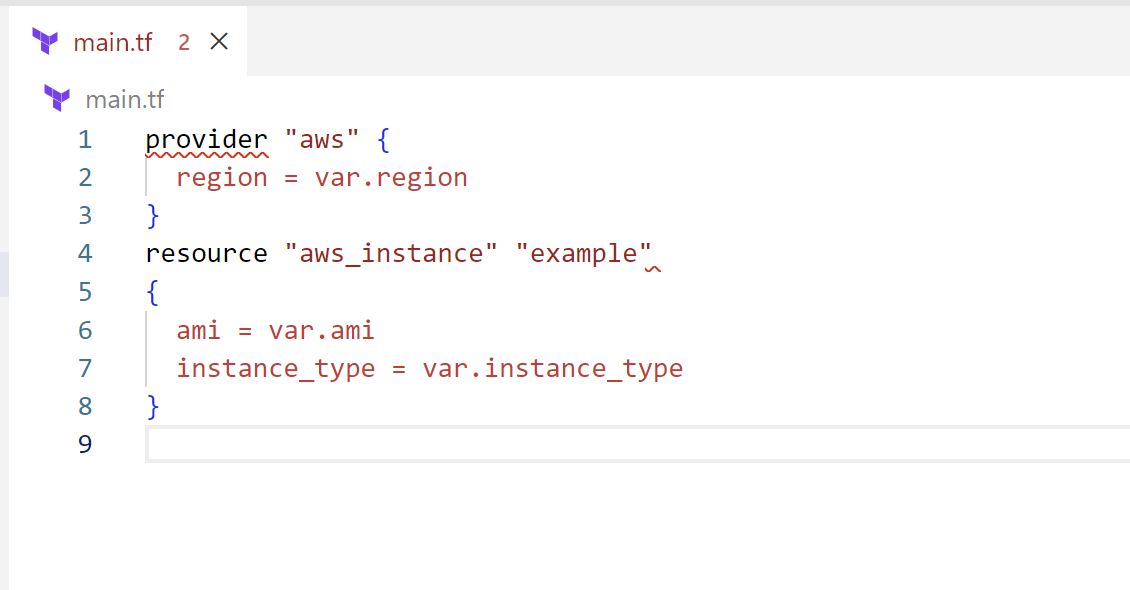
**1. Create a Terraform Directory**



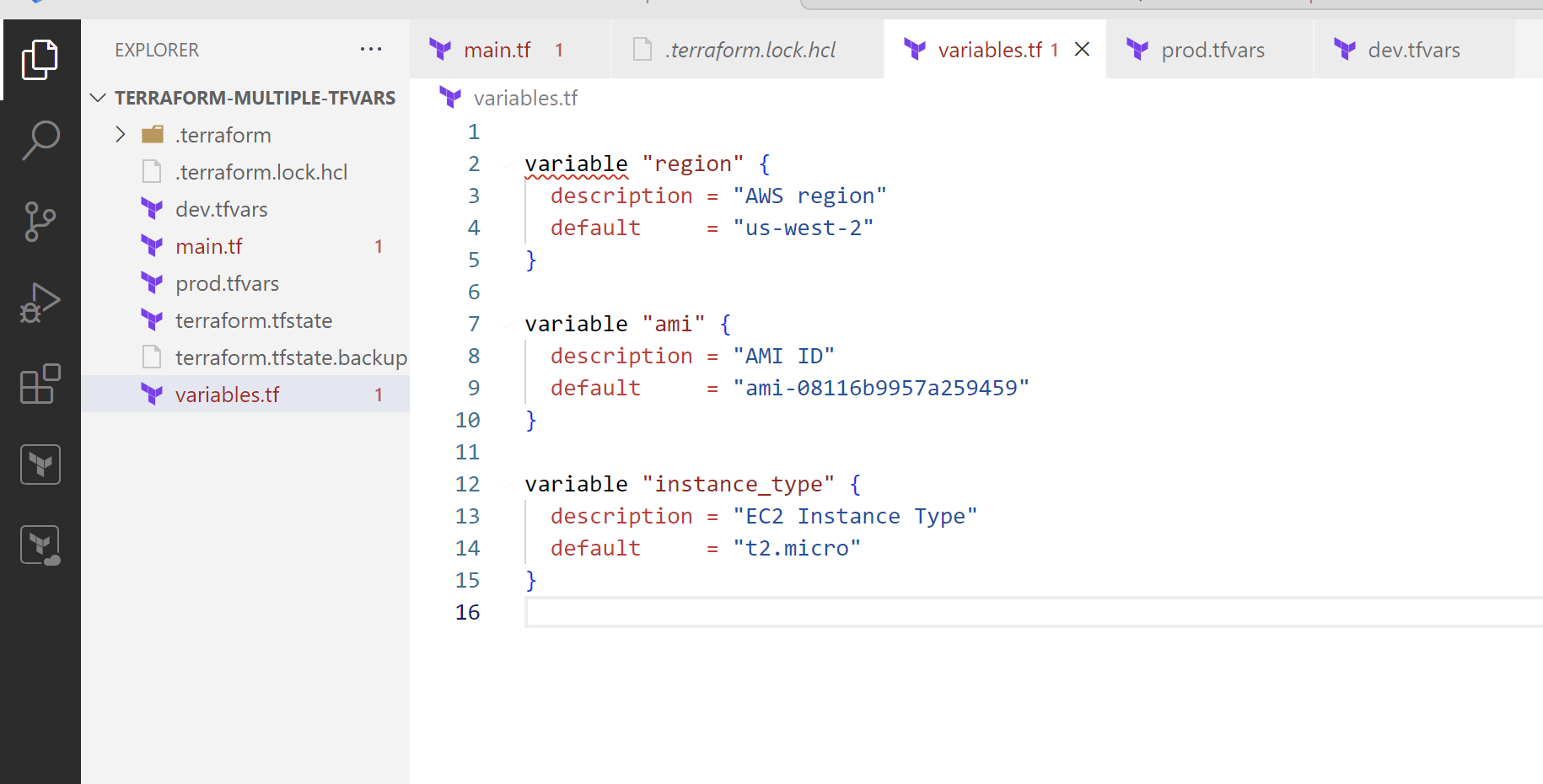
• Create Terraform Configuration Files:

• Create a file named main.tf:

# main.tf



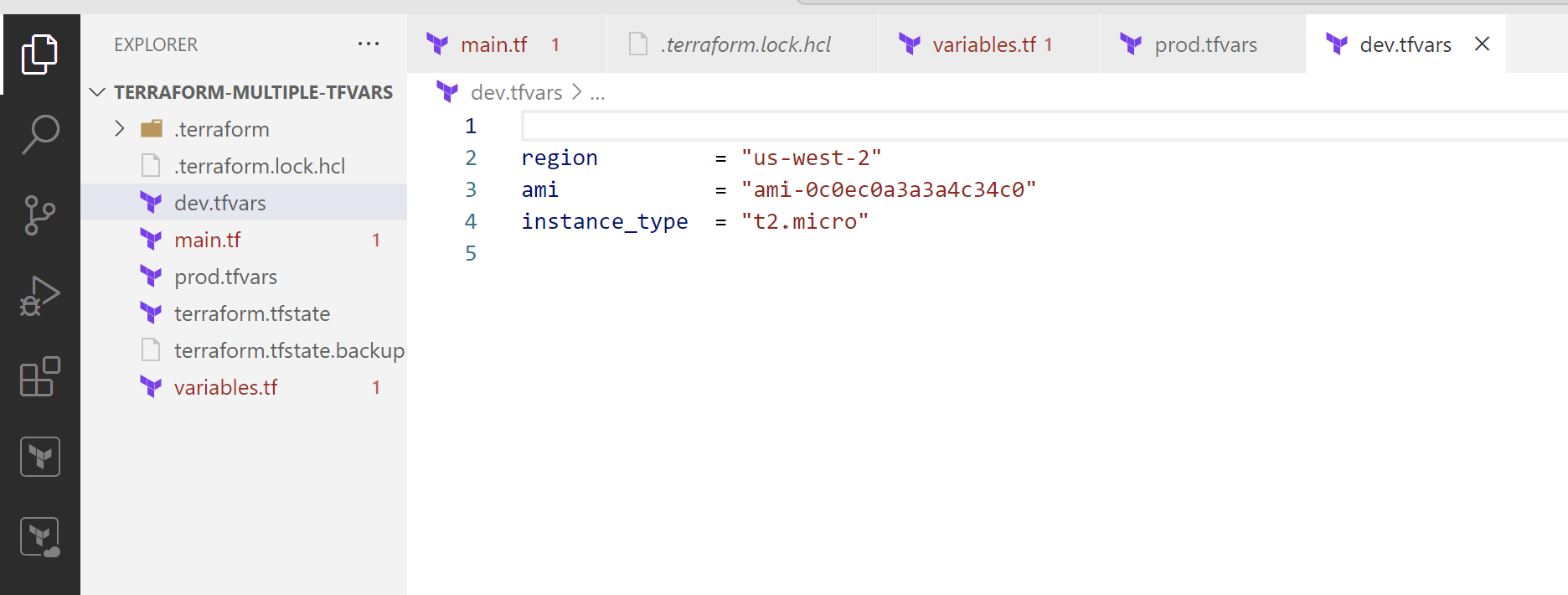
# variables.tf



2. Create Multiple tfvars Files:

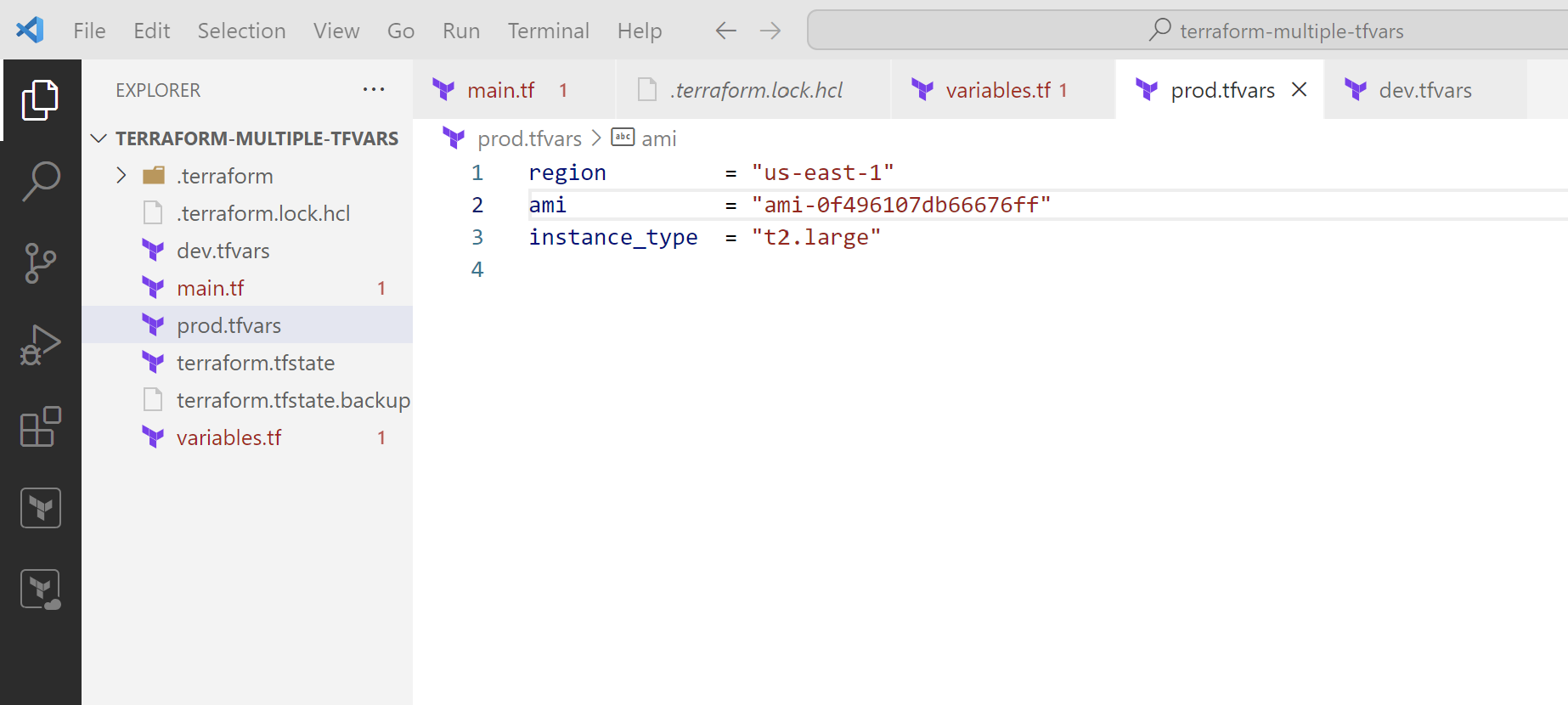
• Create a file named dev.tfvars:

# dev.tfvars



• Create a file named prod.tfvars:

# prod.tfvars

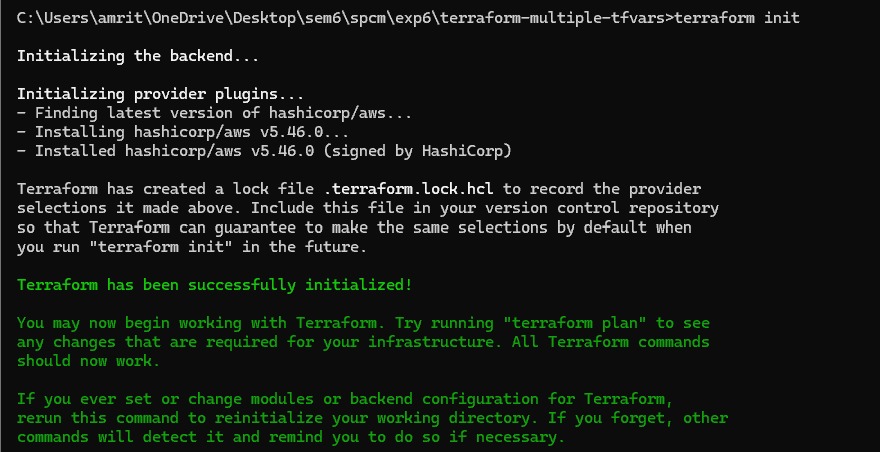


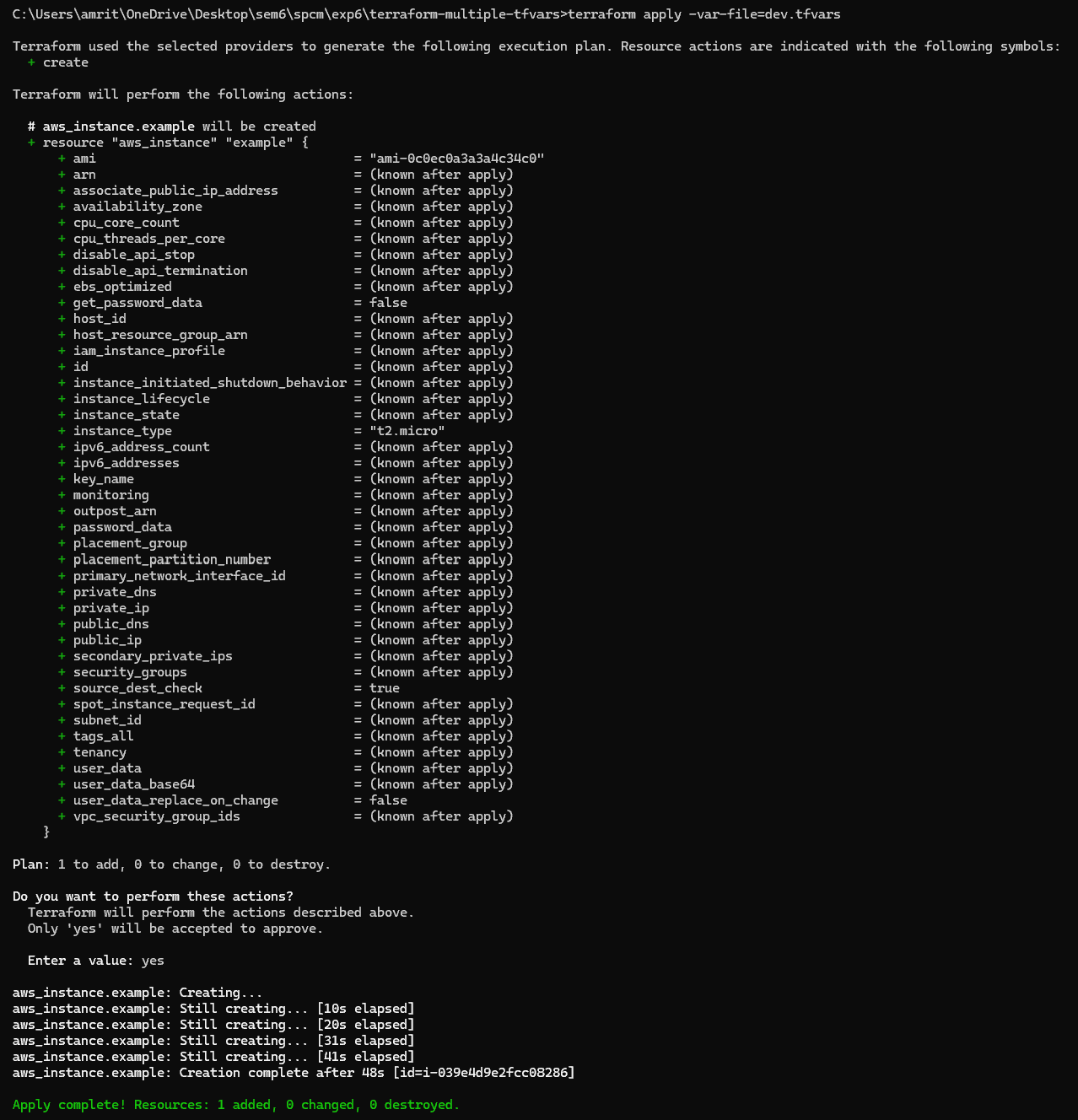
• In these files, provide values for the variables based on the environments.

3. Initialize and Apply for Dev Environment:

• Run the following Terraform commands to initialize and apply the configuration

for the dev environment:



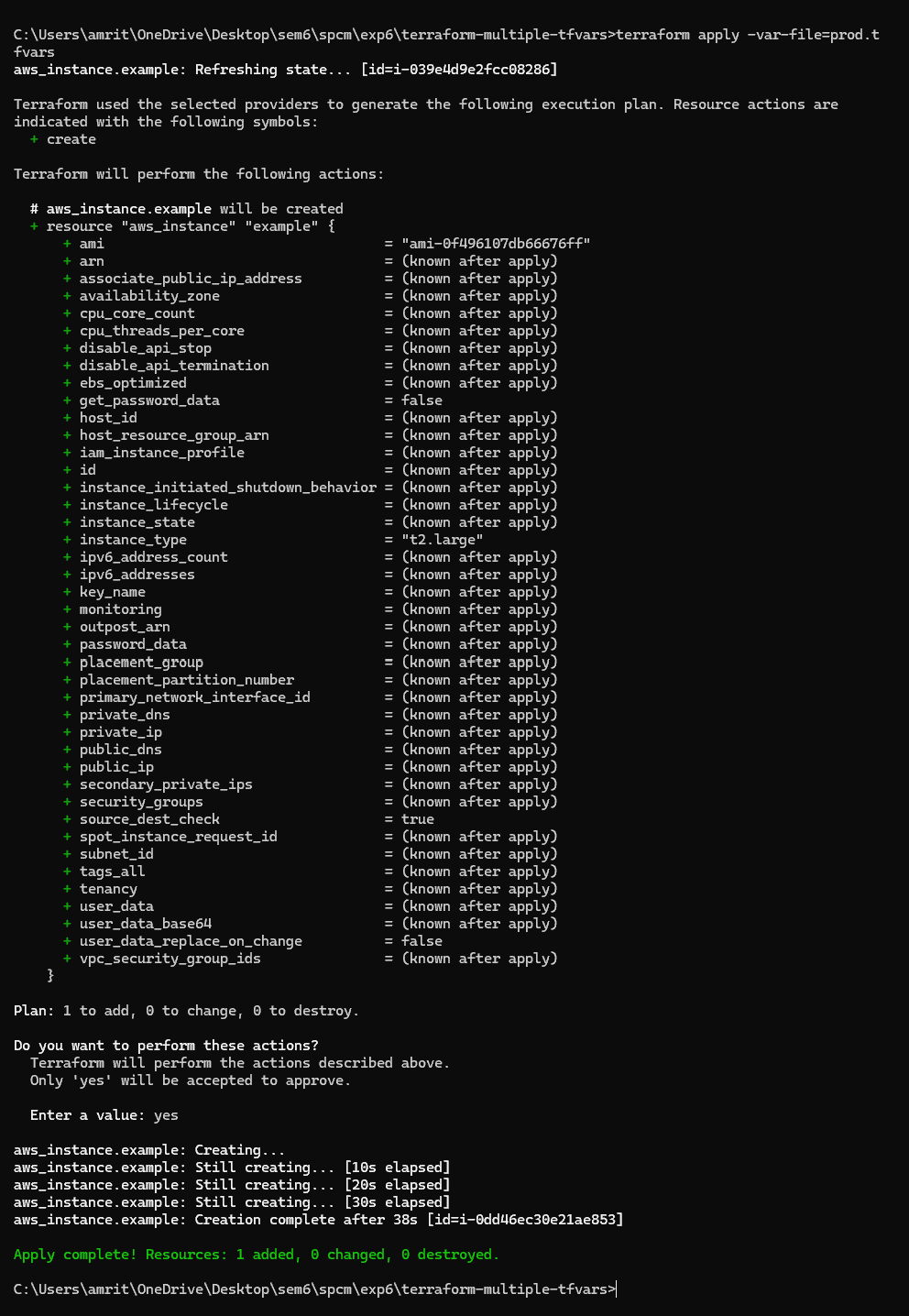


4. Initialize and Apply for Prod Environment:

• Run the following Terraform commands to initialize and apply the configuration

for the prod environment:

terraform init

terraform apply -var-file=prod.tfvars

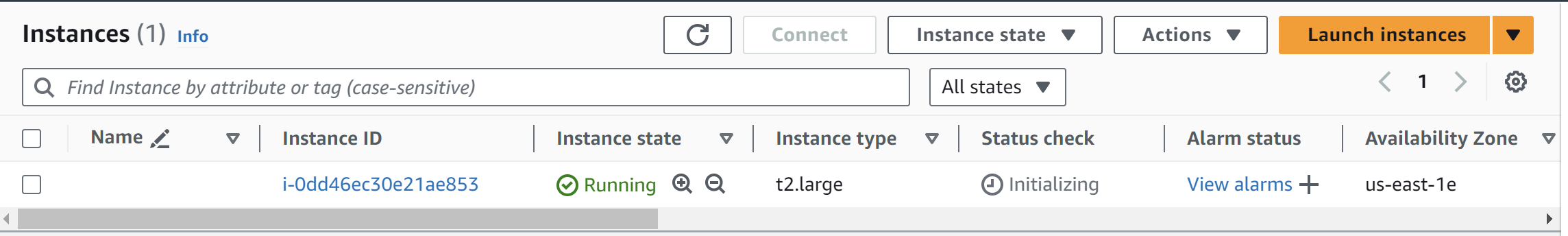
5. Test and Verify:

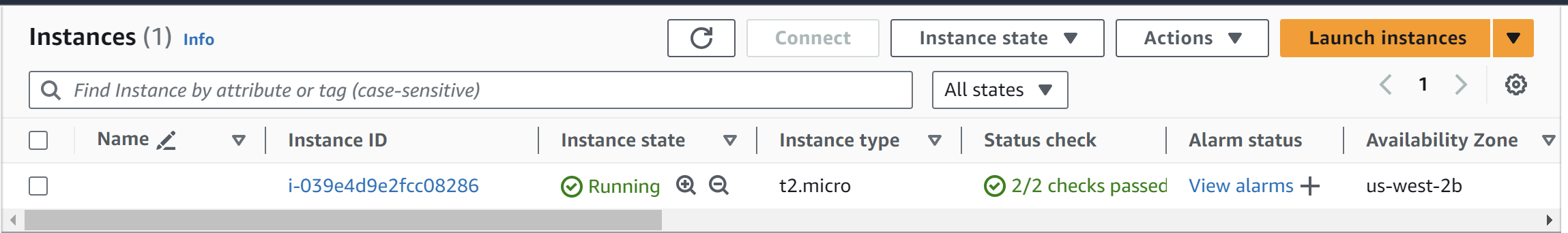
• Observe how different tfvars files are used to set variable values for different

environments during the apply process.

• Access the AWS Management Console or use the AWS CLI to verify the creation of

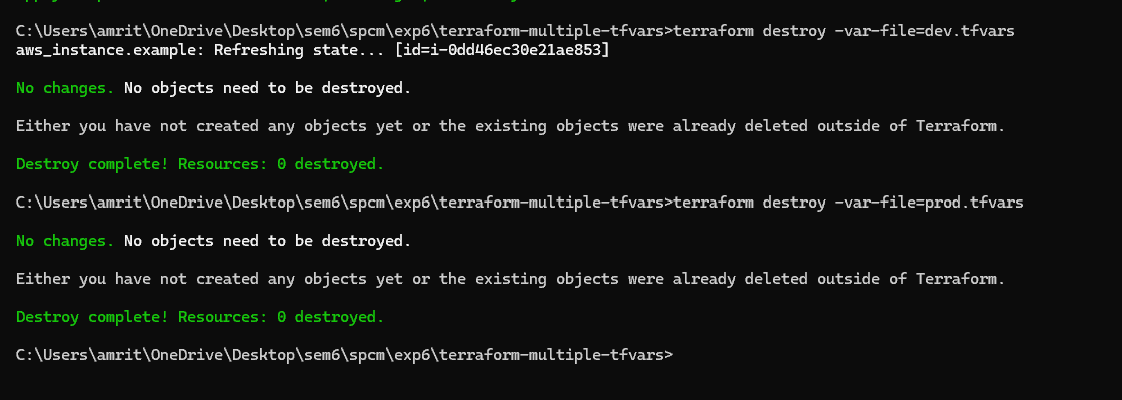
resources in the specified regions and instance types.





6. Clean Up:

• After testing, you can clean up resources:



7. Conclusion:

This lab exercise demonstrates how to use multiple tfvars files in Terraform to manage

variable values for different environments. It allows you to maintain separate

configuration files for different environments, making it easier to manage and

maintain your infrastructure code. Experiment with different values in the dev.tfvars

and prod.tfvars files to observe how they impact the infrastructure provisioning

process for each environment