🌟 Creating a Reusable Terraform Module for EC2 Instances

Terraform is a powerful tool for managing infrastructure as code. In this tutorial, we'll create a reusable module to launch EC2 instances. We'll cover the folder structure, necessary files, and how to use the module in different configurations.

📁 Folder Structure

First, let's set up the folder structure for our Terraform module:

```
$ mkdir -p modules/ec2-instance
$ cd modules/ec2-instance
```

file structure

Set Up AWS Credentials:

export AWS_ACCESS_KEY_ID="your_access_key"
export AWS_SECRET_ACCESS_KEY="your_secret_key"

```
root@ip-172-31-27-15:~/demo#
root@ip-172-31-27-15:~/demo# tree
.
├── main.tf
├── modules
│   └── ec2_instance
│       ├── main.tf
│       ├── output.tf
│       ├── terrafor.tfvars
│       └── variable.tf
└── snap

4 directories, 5 files
```

Module Configuration

The **main.tf** configuration file for the module:

main.tf 3, U

main.tf > ...

```
provider "aws" {
    region = "eu-west-3"
}



resource "aws_instance" "my-instance" {
        ami           = var.ami_value
        instance_type = var.instance_type_value

        tags = {
            Name = var.tag_id
        }
}
```

variables.tf

Define the variables for the module:

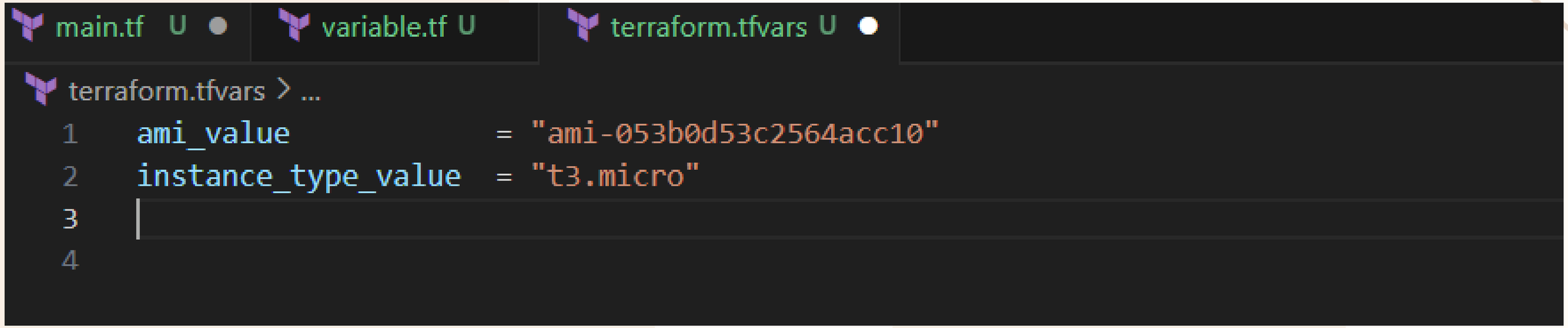
main.tf U ● variable.tf U ×

variable.tf > variable "tag_id"

```
1
2   variable "ami_value" {
3       description = "Value for the AMI"
4   }
5   variable "instance_type_value" {
6       description = "Value for the instance type"
7   }
8   variable "tag_id" {
9       description = "Tag ID to apply to the instance"
10  }
```

terraform.tfvars

Specify the values for the variables:



outputs.tf

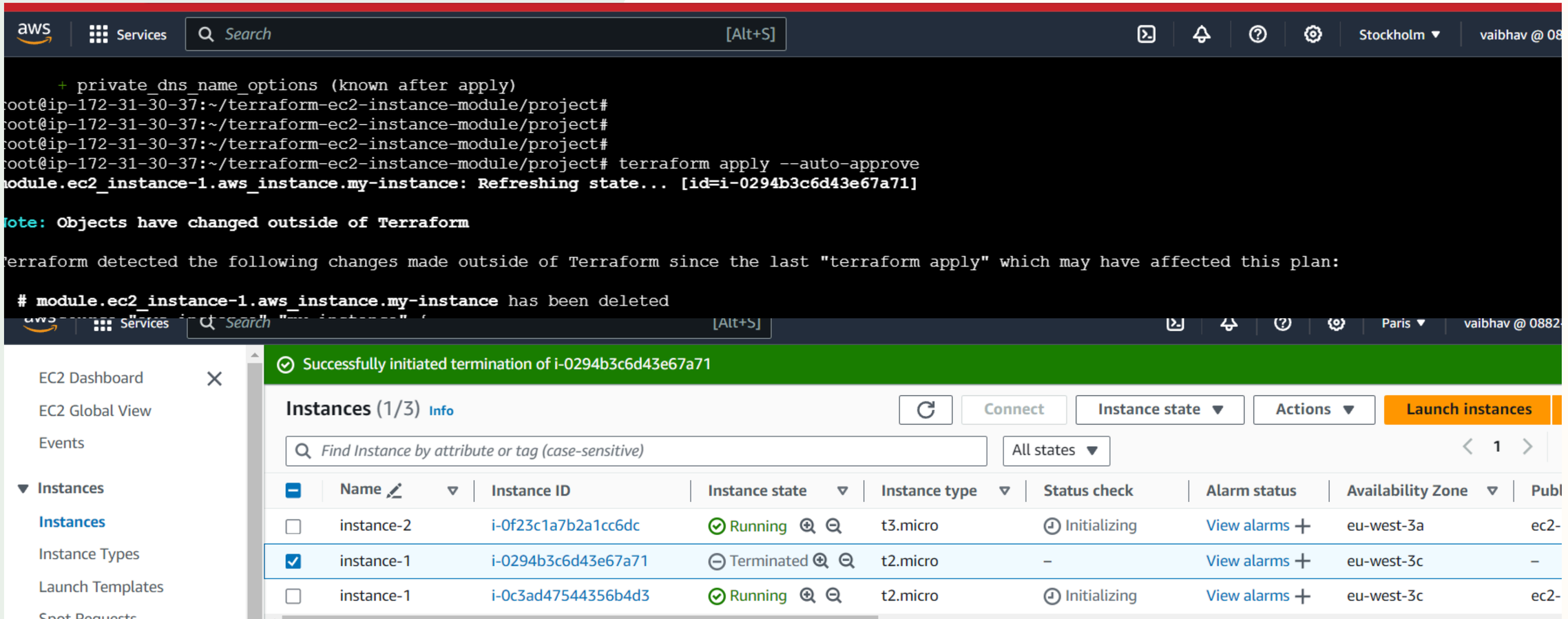
Define the outputs for the module:

Using the Module

Navigate outside of the modules/ec2-instance directory: **cd ../..**

Run Terraform Commands

cd project
terraform init
terraform apply --auto-approve



Thank you!

https://github.com/vaibhav0342/terraform-ec2-instance-module