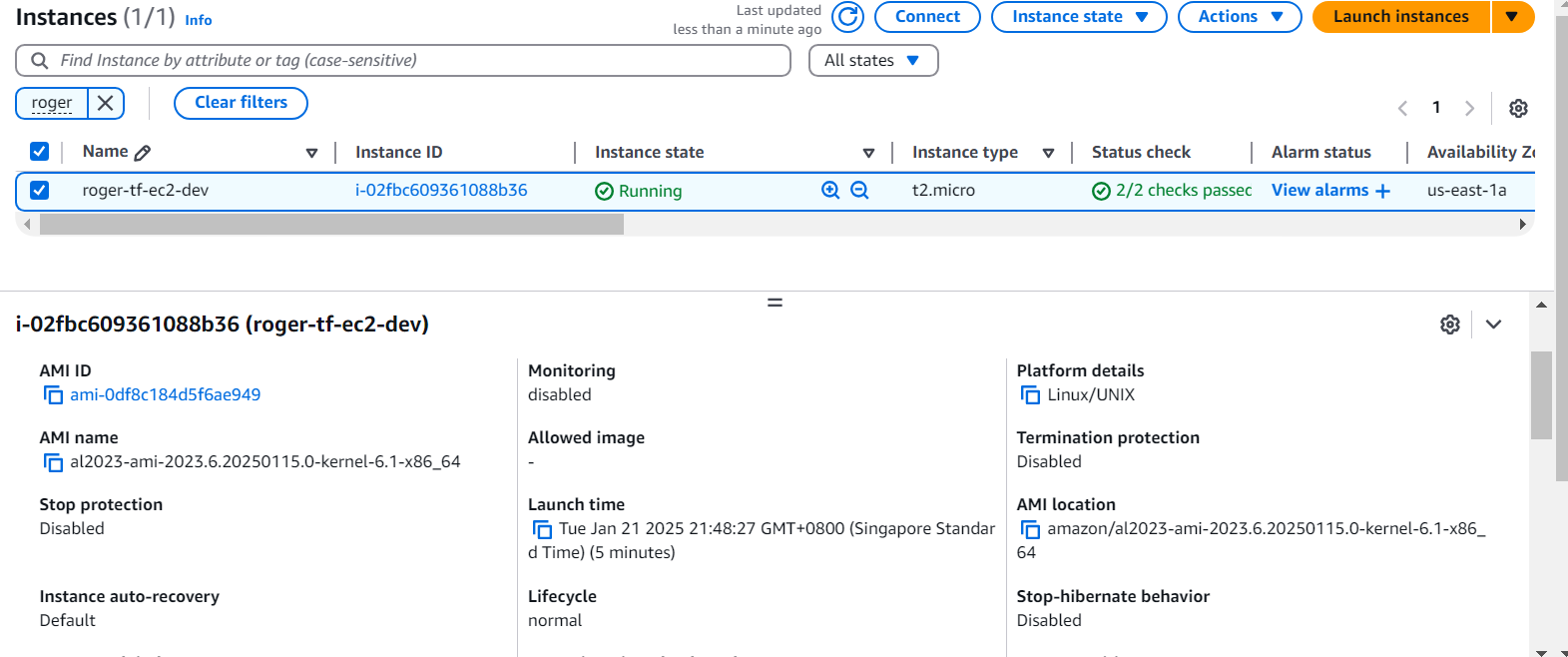
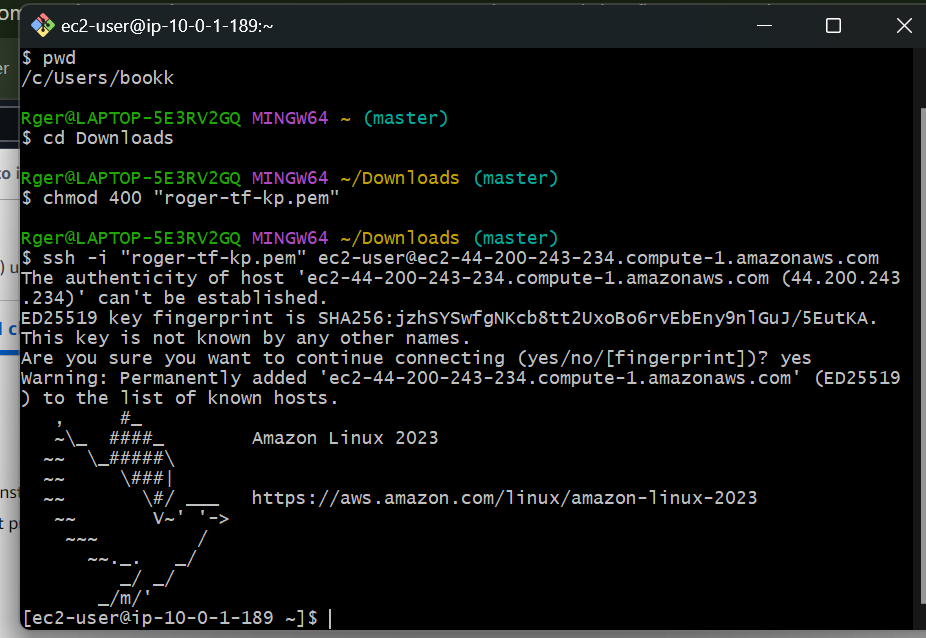
## This EC2 created from Terraform, using var.tf (attached behind the steps for reference)

## 





## Activities

Update with the following files:

* main.tf / compute.tf

locals {

resource\_prefix = "luqman-tf"

}

resource "aws\_instance" "public" {

ami = "ami-0df8c184d5f6ae949" #Challenge, find the AMI ID of Amazon Linux 2 in us-east-1

instance\_type = "t2.micro"

subnet\_id = data.aws\_subnet.existing\_ce9\_pub\_subnet.id

associate\_public\_ip\_address = true

key\_name = "luqman-test-keypair" #Change to your keyname, e.g. jazeel-key-pair

vpc\_security\_group\_ids = [aws\_security\_group.allow\_ssh.id]

tags = {

Name = "${ local.resource\_prefix }-ec2-${ var.env }" # Ensure your

}

}

output "ec2\_public\_ip" {

value = aws\_instance.public.public\_ip

}

resource "aws\_security\_group" "allow\_ssh" {

name = "${ local.resource\_prefix }-security-group-ssh"

description = "Allow SSH inbound"

vpc\_id = data.aws\_vpc.existing\_ce9\_vpc.id

}

resource "aws\_vpc\_security\_group\_ingress\_rule" "allow\_tls\_ipv4" {

security\_group\_id = aws\_security\_group.allow\_ssh.id

cidr\_ipv4 = "0.0.0.0/0"

from\_port = 22

ip\_protocol = "tcp"

to\_port = 22

}

* vpc.tf

# These blocks are for filtering existing VPC and Public Subnet

data "aws\_vpc" "existing\_ce9\_vpc" {

filter {

name = "tag:Name"

values = ["vpc\_ce9\_learner"]

}

}

data "aws\_subnet" "existing\_ce9\_pub\_subnet" {

filter {

name = "tag:Name"

values = ["subnet\_ce9\_learner\_us-east-1a"]

}

}

* variables.tf

variable "env" {

description = "The environment of the AWS infrastructure"

type = string

default = "dev"

}

variable "vpc\_name" {

description = "The VPC Name to use"

type = string

default = "vpc\_ce9\_learner"

}

variable "subnet\_name" {

description = "The VPC Name to use"

type = string

default = "subnet\_ce9\_learner\_us-east-1a"

}

* provider.tf

terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "5.83.1"

}

}

}

provider "aws" {

region = "us-east-1"

}

* backend.tf

terraform {

backend "s3" {

bucket = "sctp-ce9-tfstate"

key = "<NAME>-ce9-module2-lesson3.tfstate" # Replace the value of key to <your suggested name>.tfstat

region = "us-east-1"

}

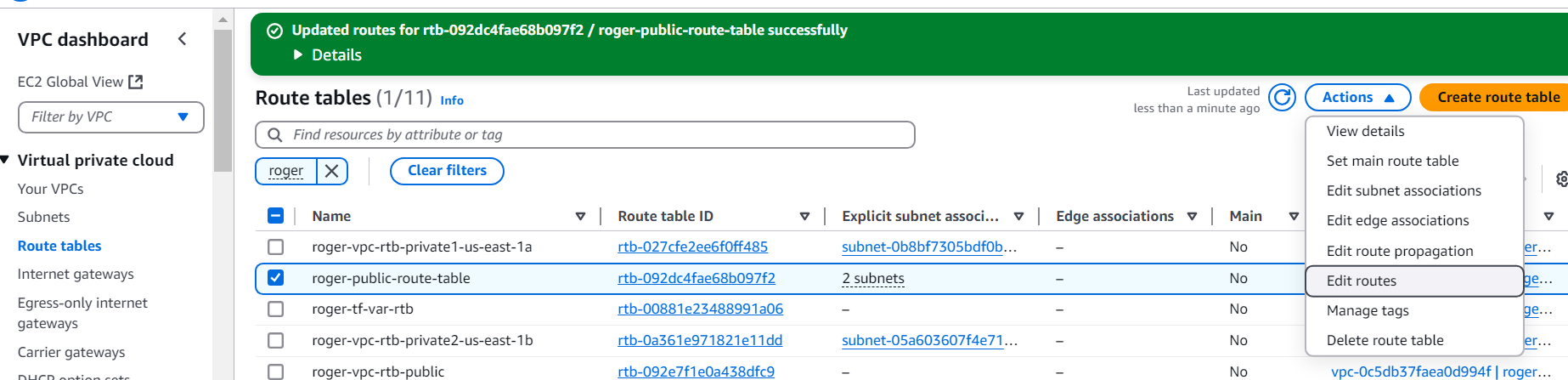
}

Challenges:

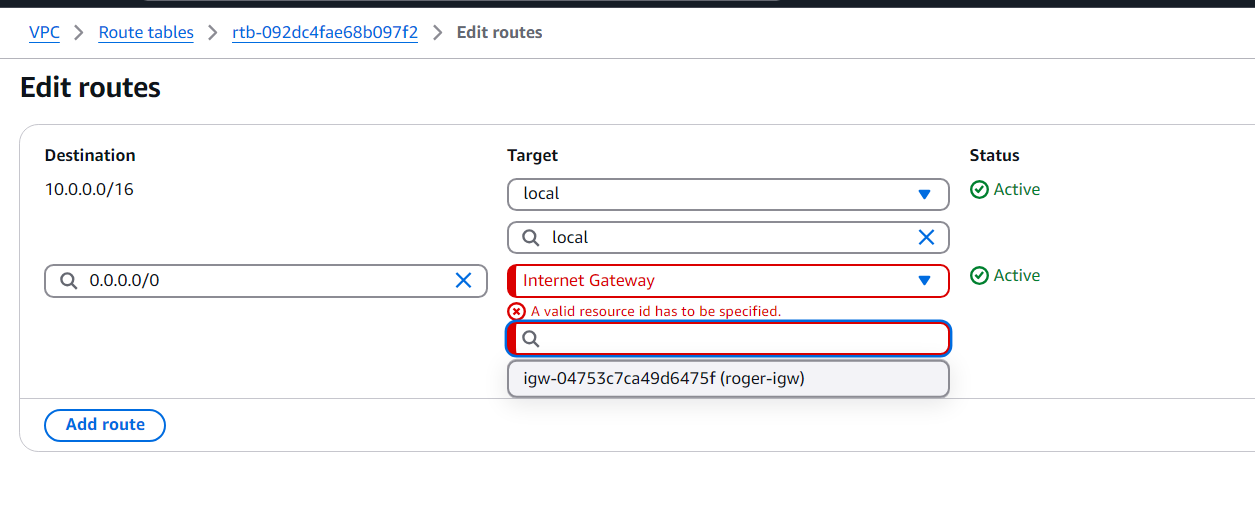
1. Create your own VPC, Subnets, IGW, Route Tables etc. Remember to delete your existing VPC before creating a new one.

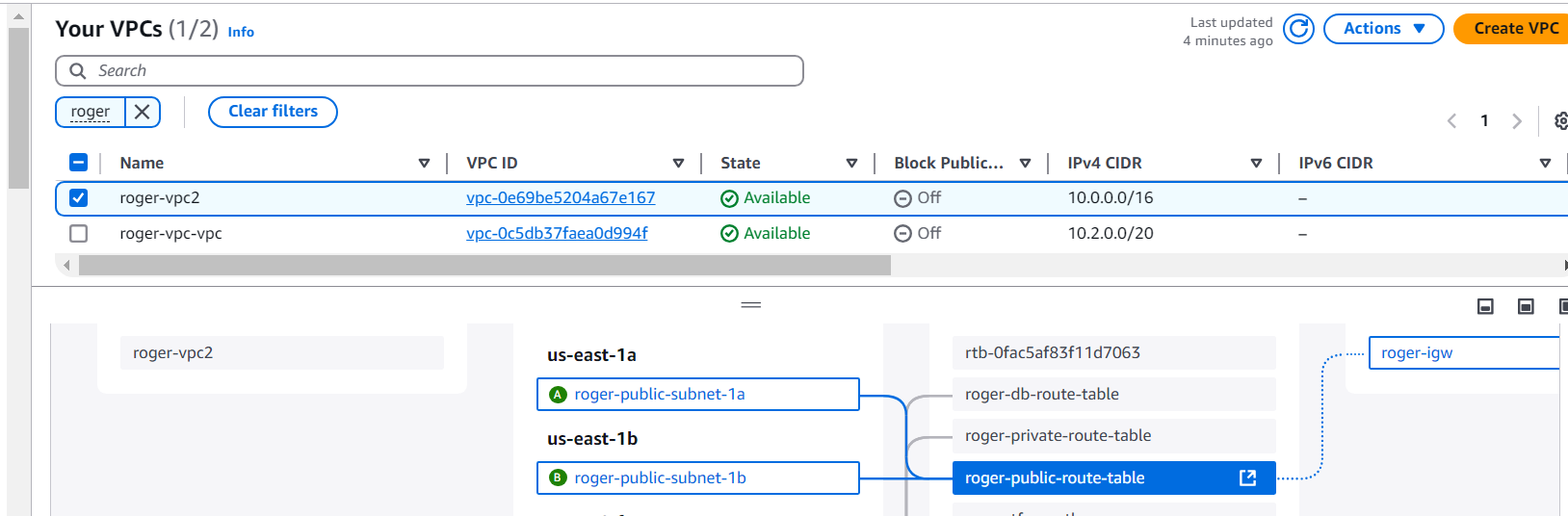
Also check route-tables to internet gateway for those subnets you use

If not, just recreate another IGW to attached to your VPC.

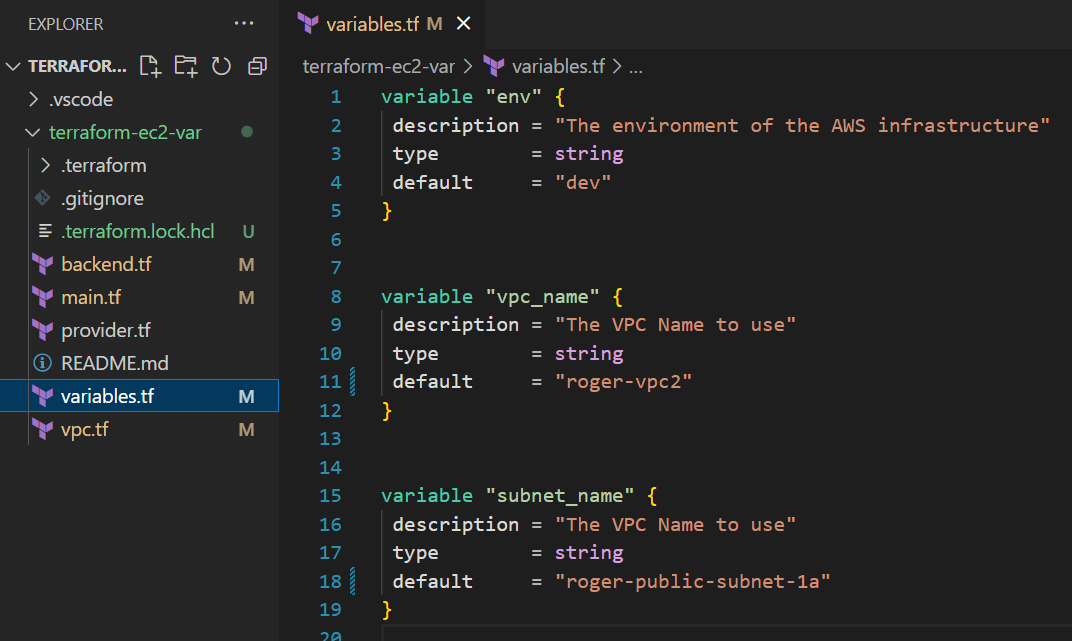


(refer to notes on VPC | IGW | attach VPC to IGW > VPC > Route table > attached to available VPC)





1. Convert your static values such as EC2 instance type, key pair name, cidr range etc. to variables rather than hardcoded values.



1. Filter for AWS AMI ID (Amazon Linux 2 AMI) using <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/data-sources/ami> instead of hardcoding the value.

Final Challenge:

1. Recall the launch template + ASG + Load Balancer activity that we did manually from the console in module 1. Recreate all those using Terraform resources :)