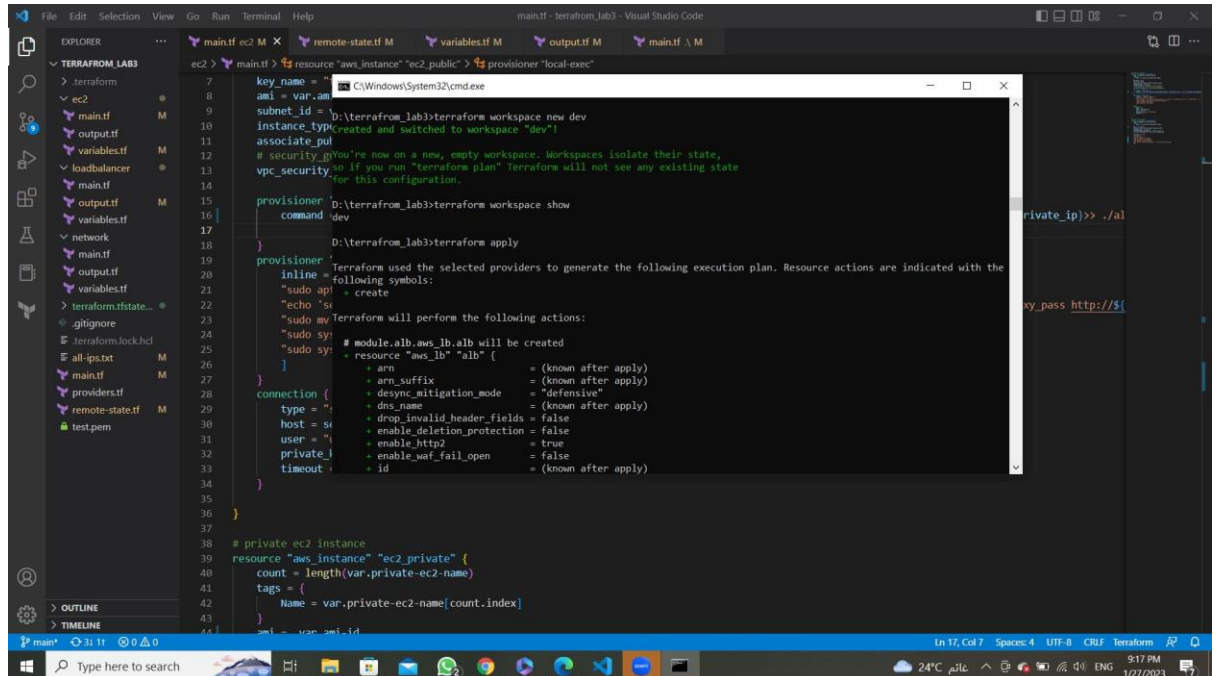


Marwa Ahmed Darwish

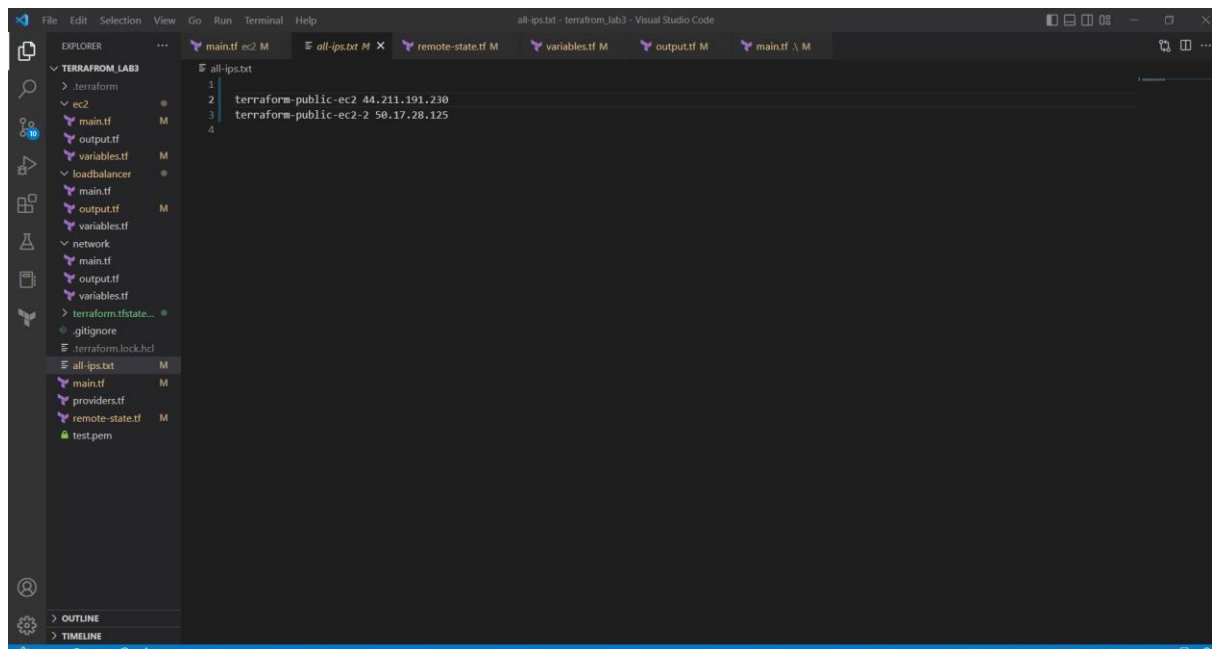
Lab 3 :

## 1- New Workspace



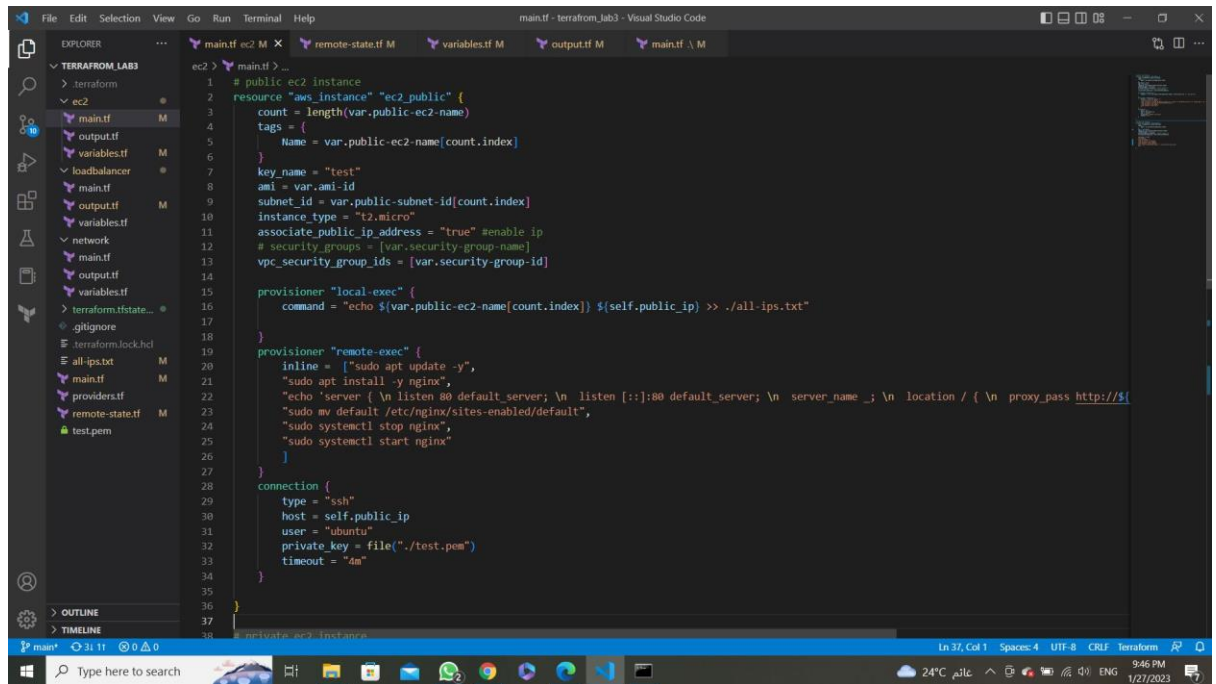
```
key_name = "C:\Windows\System32\cmd.exe"
ami = var.ami
subnet_id = "d:\terraform_lab3\terraform workspace new dev"
instance_type = "t2.micro"
associate_public_ip_address = true
# security group
vpc_security_group_ids = ["sg-01234567"]
provisioner "local-exec" {
  command = "D:\terraform_lab3\terraform workspace show"
}
provisioner "local-exec" {
  command = "D:\terraform_lab3\terraform apply"
}
provisioner "local-exec" {
  inline = [
    "terraform init",
    "terraform plan",
    "terraform apply"
  ]
}
# module.alb.aws_lb.alb will be created
resource "aws_lb" "alb" {
  name               = "alb"
  arn                = (known after apply)
  arn_suffix         = (known after apply)
  desync_mitigation_mode = "defensive"
  dns_name           = (known after apply)
  drop_invalid_header_fields = false
  enable_deletion_protection = false
  enable_http2       = true
  enable_waf_fail_open = false
  id                 = (known after apply)
}
connection {
  type = "ssh"
  host = "10.0.0.1"
  user = "ubuntu"
  private_key_path = "test.pem"
  timeout = 30
}
# private ec2 instance
resource "aws_instance" "ec2_private" {
  count = length(var.private-ec2-name)
  tags = {
    Name = var.private-ec2-name[count.index]
  }
  ami = var.ami
}
```

## 2- Public ip of public ec2



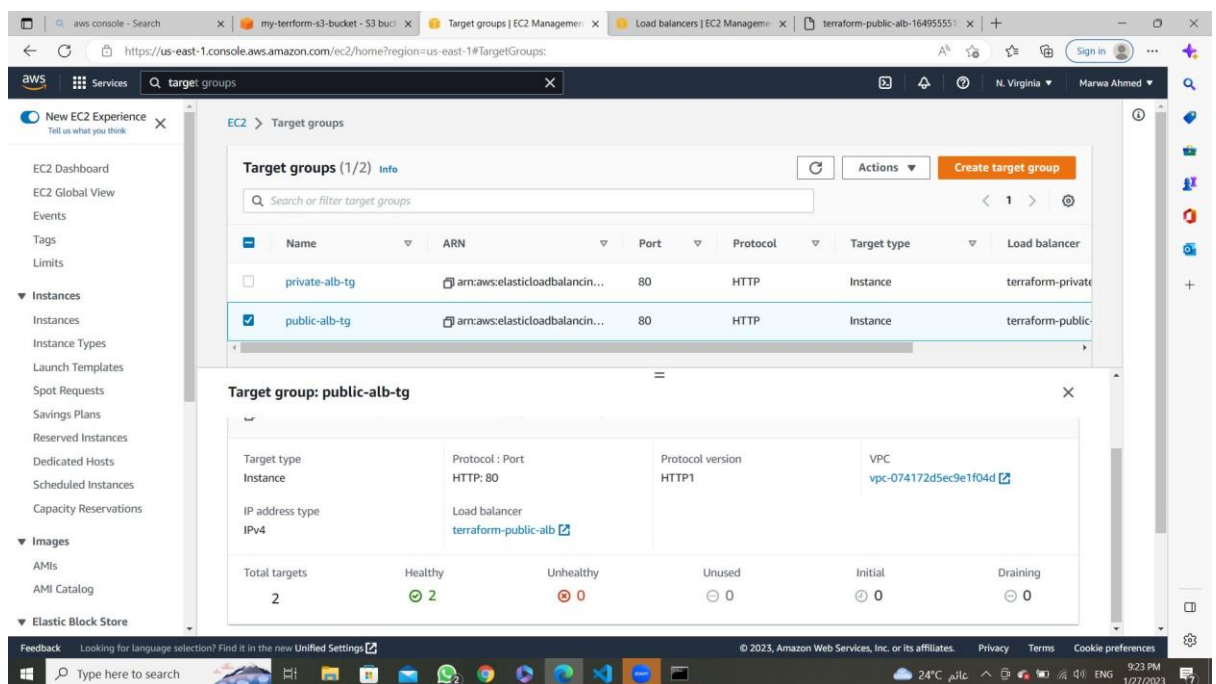
```
1
2 terraform-public-ec2 44.211.191.230
3 terraform-public-ec2-2 50.17.28.125
4
```

### 3- Configuration of proxy



```
1 # public ec2 instance
2 resource "aws_instance" "ec2_public" {
3   count = length(var.public-ec2-name)
4   tags = {
5     Name = var.public-ec2-name[count.index]
6   }
7   key_name = "test"
8   ami = var.ami-id
9   subnet_id = var.public-subnet-id[count.index]
10  instance_type = "t2.micro"
11  associate_public_ip_address = "true" #enable ip
12  # security_groups = [var.security-group-name]
13  vpc_security_group_ids = [var.security-group-id]
14
15  provisioner "local-exec" {
16    command = "echo ${var.public-ec2-name[count.index]} ${self.public_ip} >> ./all-ips.txt"
17  }
18
19  provisioner "remote-exec" {
20    inline = [
21      "sudo apt update -y",
22      "sudo apt install -y nginx",
23      "echo 'server { \n listen 80 default_server; \n listen [::]:80 default_server; \n server_name _; \n location / { \n proxy_pass http://$",
24      "sudo mv default /etc/nginx/sites-enabled/default",
25      "sudo systemctl stop nginx",
26      "sudo systemctl start nginx"
27    ]
28  }
29
30  connection {
31    type = "ssh"
32    host = self.public_ip
33    user = "ubuntu"
34    private_key = file("../test.pem")
35    timeout = "4m"
36  }
37 }
```

### 4- Public load balancer target group



The screenshot shows the AWS Management Console interface for the 'Target groups' section. The 'public-alb-tg' target group is selected, and its configuration details are displayed in a modal window.

Name	ARN	Port	Protocol	Target type	Load balancer
private-alb-tg	arn:aws:elasticloadbalancing:us-east-1:16495551:targetgroup/private-alb-tg/...	80	HTTP	Instance	terraform-private-alb
public-alb-tg	arn:aws:elasticloadbalancing:us-east-1:16495551:targetgroup/public-alb-tg/...	80	HTTP	Instance	terraform-public-alb

Target group: public-alb-tg					
Target type	Protocol : Port	Protocol version	VPC		
Instance	HTTP: 80	HTTP1	vpc-074172d5ec9e1f04d		
IP address type	Load balancer	terraform-public-alb			
IPv4					
Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0

## 5- Private load balancer target group

The screenshot shows the AWS Management Console 'Target groups' page. The 'private-alb-tg' target group is selected, showing its details and registered targets.

**Target groups (1/2)**

Name	ARN	Port	Protocol	Target type	Load balancer
private-alb-tg	arn:aws:elasticloadbalancing:us-east-1:16495551:targetgroup/private-alb-tg/...	80	HTTP	Instance	terraform-private-alb
public-alb-tg	arn:aws:elasticloadbalancing:us-east-1:16495551:targetgroup/public-alb-tg/...	80	HTTP	Instance	terraform-public-alb

**Target group: private-alb-tg**

**Registered targets (2)**

Instance ID	Name	Port	Zone	Health status	Health status details
i-0a6aae101d8903b5d	terraform-private-ec2-2	80	us-east-1b	healthy	
i-03771a7139712b239	terraform-private-ec2	80	us-east-1a	healthy	

## 6- Private and public Load balancer

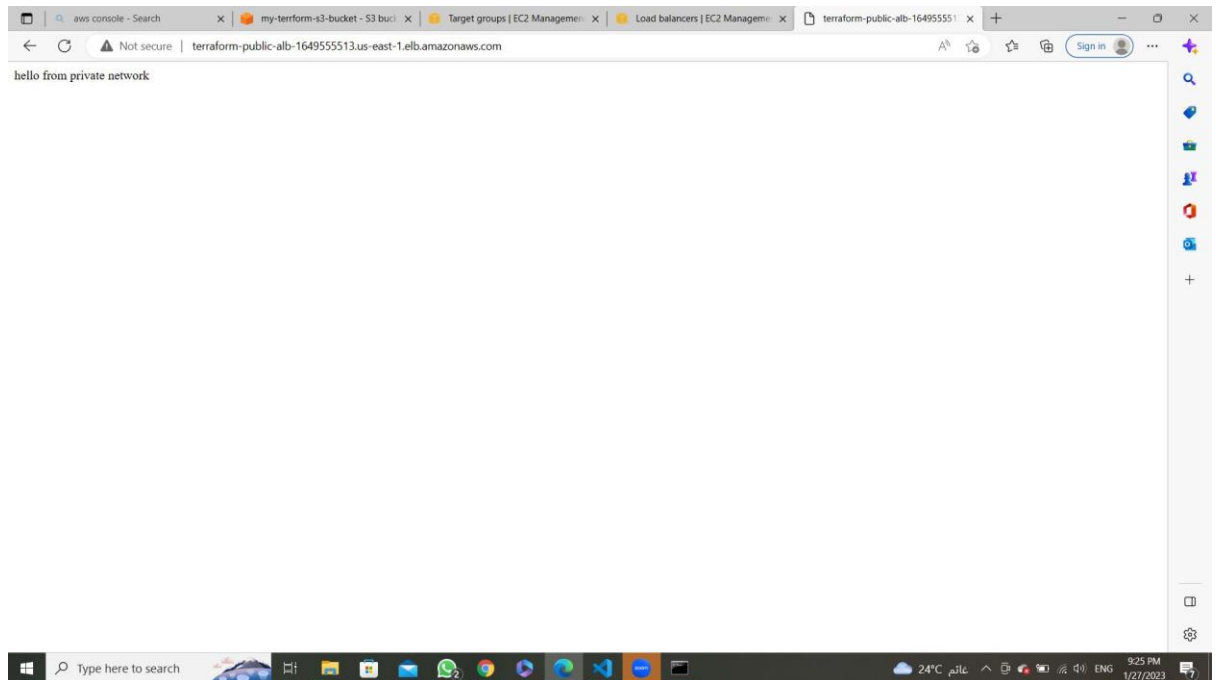
The screenshot shows the AWS Management Console 'Load balancers' page. The 'terraform-public-alb' and 'terraform-private-alb' load balancers are listed.

**Load balancers (1/2)**

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type
terraform-public-alb	terraform-public-alb-1649...	Active	vpc-074172d5ec9e1f04d	2 Availability Zones	application
terraform-private-alb	internal-terraform-private-...	Active	vpc-074172d5ec9e1f04d	2 Availability Zones	application

## 7- Public dns name in browser



## 8- S3 bucket

