Terraform_VPC_EC2

Для початку перевіримо версію встановлених раніше terraform та CLI на машині

```
dmytro@ubuntuserver:~$ terraform -version
Terraform v1.9.8
on linux_amd64
dmytro@ubuntuserver:~$
```

Створимо структуру (директорію) проекту – *aws-vpc-exercise* із наступними файлами

```
-rw-rw-r-- 1 dmytro dmytro 2733 Nov 14 13:57 main.tf
-rw-rw-r-- 1 dmytro dmytro  318 Nov 14 13:57 outputs.tf
-rw-rw-r-- 1 dmytro dmytro  693 Nov 14 13:57 variables.tf
```

Вміст файлу variables.tf (змінні для спрощення налаштувань конфігурації)

```
variable "aws_region" {
  description = "AWS region for deployment"
  default     = "us-east-1"
}

variable "vpc_cidr" {
  description = "CIDR block for VPC"
  default     = "10.0.0.0/16"
}

variable "public_subnet_cidr" {
  description = "CIDR block for public subnet"
  default     = "10.0.1.0/24"
}

variable "private_subnet_cidr" {
  description = "CIDR block for private subnet"
  default     = "10.0.2.0/24"
}

# Optional: Add an output for easier debugging
output "vpc_cidr" {
  value = var.vpc_cidr
}

output "public_subnet_cidr" {
  value = var.public_subnet_cidr
}

output "private_subnet_cidr" {
  value = var.private_subnet_cidr
}

output "aws_region" {
  value = var.aws_region
}

```

Вміст файлу outputs.tf (вивід даних, що будуть показуватись після terraform apply)

```
output "public_instance_id" {
  value = aws_instance.public_instance.id
}

output "public_instance_ip" {
  value = aws_instance.public_instance.public_ip
}

output "private_instance_id" {
  value = aws_instance.private_instance.id
}

output "private_instance_ip" {
  value = aws_instance.private_instance.private_ip
}

```

Вміст файлу main.tf (основні налаштування VPC, підмереж та EC2 інстансів) – створює VPC, публічну та приватну мережі, інтернет шлюз, таблицю маршрутизації та EC2 інстанси

```
provider "aws" {
  region = var.aws_region
}

resource "aws_vpc" "my_vpc" {
  cidr_block           = var.vpc_cidr
  enable_dns_support   = true
  enable_dns_hostnames = true
  tags = {
    Name = "MyVPC"
  }
}

resource "aws_subnet" "public_subnet" {
  vpc_id                  = aws_vpc.my_vpc.id
  cidr_block              = var.public_subnet_cidr
  map_public_ip_on_launch = true
  tags = {
    Name = "PublicSubnet"
  }
}

resource "aws_subnet" "private_subnet" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = var.private_subnet_cidr
  tags = {
    Name = "PrivateSubnet"
  }
}

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.my_vpc.id
  tags = {
    Name = "InternetGateway"
  }
}

resource "aws_route_table" "public_rt" {
  vpc_id = aws_vpc.my_vpc.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }
  tags = {
    Name = "PublicRouteTable"
  }
}

resource "aws_route_table_association" "public_rt_association" {
  subnet_id      = aws_subnet.public_subnet.id
  route_table_id = aws_route_table.public_rt.id
}
```

```
 resource "aws_security_group" "public_sg" {
   vpc_id = aws_vpc.my_vpc.id

   ingress {
     from_port   = 22
     to_port     = 22
     protocol    = "tcp"
     cidr_blocks = ["0.0.0.0/0"]
   }

   ingress {
     from_port   = 80
     to_port     = 80
     protocol    = "tcp"
     cidr_blocks = ["0.0.0.0/0"]
   }

   egress {
     from_port   = 0
     to_port     = 0
     protocol    = "-1"
     cidr_blocks = ["0.0.0.0/0"]
   }

   tags = {
     Name = "PublicSG"
   }
 }

 resource "aws_security_group" "private_sg" {
   vpc_id = aws_vpc.my_vpc.id

   ingress {
     from_port       = 22
     to_port         = 22
     protocol        = "tcp"
     security_groups = [aws_security_group.public_sg.id]
   }

   egress {
     from_port   = 0
     to_port     = 0
     protocol    = "-1"
     cidr_blocks = ["0.0.0.0/0"]
   }

   tags = {
     Name = "PrivateSG"
   }
 }
```

```
resource "aws_instance" "public_instance" {
  ami                         = "ami-0ed6534c7d6a8e78f"
  instance_type               = "t2.micro"
  subnet_id                   = aws_subnet.public_subnet.id
  vpc_security_group_ids      = [aws_security_group.public_sg.id]
  associate_public_ip_address = true  # Додано для публічної IP-адреси

  tags = {
    Name = "PublicInstance"
  }
}

resource "aws_instance" "private_instance" {
  ami                    = "ami-0ed6534c7d6a8e78f"
  instance_type          = "t2.micro"
  subnet_id              = aws_subnet.private_subnet.id
  vpc_security_group_ids = [aws_security_group.private_sg.id]

  tags = {
    Name = "PrivateInstance"
  }
}

```

Ініціалізація Terraform

```
dmytro@ubuntuserver:~/dan_it_homeworks/aws-vpc-exercise$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.75.1...
- Installed hashicorp/aws v5.75.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

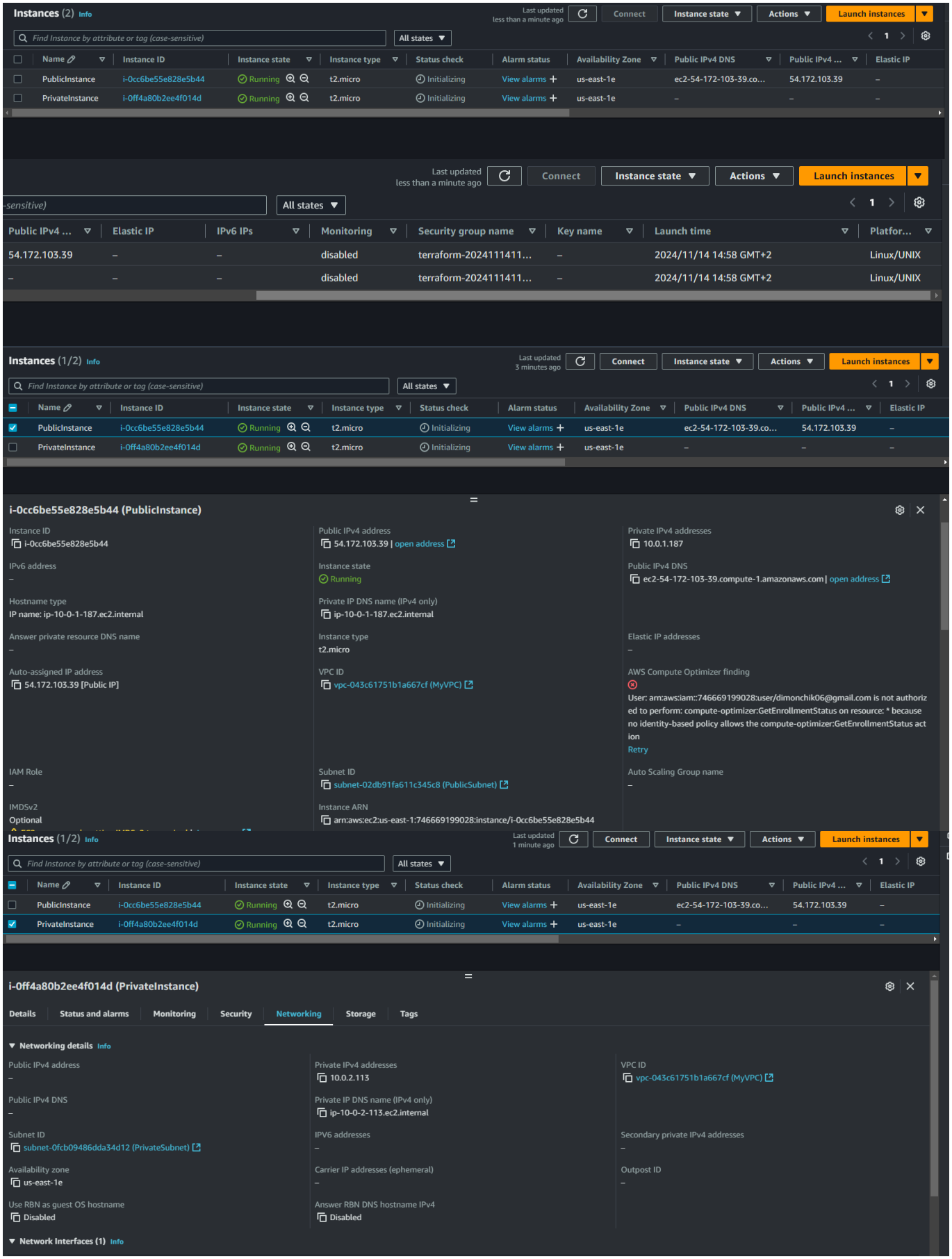
Перевірка плану

```
Plan: 10 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + aws_region          = "us-east-1"
  + private_instance_id = (known after apply)
  + private_instance_ip = (known after apply)
  + private_subnet_cidr = "10.0.2.0/24"
  + public_instance_id  = (known after apply)
  + public_instance_ip  = (known after apply)
  + public_subnet_cidr  = "10.0.1.0/24"
  + vpc_cidr            = "10.0.0.0/16"
```

Застосування конфігурації

```
Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + private_instance_id = (known after apply)
  + private_instance_ip = (known after apply)
  + public_instance_id  = (known after apply)
  + public_instance_ip  = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.public_instance: Creating...
aws_instance.private_instance: Creating...
aws_instance.public_instance: Still creating... [10s elapsed]
aws_instance.private_instance: Still creating... [10s elapsed]
aws_instance.public_instance: Still creating... [20s elapsed]
aws_instance.private_instance: Still creating... [20s elapsed]
aws_instance.public_instance: Still creating... [30s elapsed]
aws_instance.private_instance: Still creating... [30s elapsed]
aws_instance.public_instance: Still creating... [40s elapsed]
aws_instance.private_instance: Still creating... [40s elapsed]
aws_instance.private_instance: Creation complete after 46s [id=i-0ff4a80b2ee4f014d]
aws_instance.public_instance: Still creating... [50s elapsed]
aws_instance.public_instance: Still creating... [1m0s elapsed]
aws_instance.public_instance: Creation complete after 1m6s [id=i-0cc6be55e828e5b44]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

aws_region = "us-east-1"
private_instance_id = "i-0ff4a80b2ee4f014d"
private_instance_ip = "10.0.2.113"
private_subnet_cidr = "10.0.2.0/24"
public_instance_id = "i-0cc6be55e828e5b44"
public_instance_ip = "54.172.103.39"
public_subnet_cidr = "10.0.1.0/24"
vpc_cidr = "10.0.0.0/16"
dmytro@ubuntuserver:~/dan_it_homeworks/aws-vpc-exercise$
```

Перевірка інстансів



Видалення конфігурації (terraform destroy)

```
Plan: 0 to add, 0 to change, 10 to destroy.

Changes to Outputs:
  - aws_region          = "us-east-1" -> null
  - private_instance_id = "i-0ff4a80b2ee4f014d" -> null
  - private_instance_ip = "10.0.2.113" -> null
  - private_subnet_cidr = "10.0.2.0/24" -> null
  - public_instance_id  = "i-0cc6be55e828e5b44" -> null
  - public_instance_ip  = "54.172.103.39" -> null
  - public_subnet_cidr  = "10.0.1.0/24" -> null
  - vpc_cidr            = "10.0.0.0/16" -> null

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_route_table_association.public_rt_association: Destroying... [id=rtbassoc-0e1c13fce84440461]
aws_instance.public_instance: Destroying... [id=i-0cc6be55e828e5b44]
aws_instance.private_instance: Destroying... [id=i-0ff4a80b2ee4f014d]
aws_route_table_association.public_rt_association: Destruction complete after 3s
aws_route_table.public_rt: Destroying... [id=rtb-0ff7bc3eb811917c9]
aws_route_table.public_rt: Destruction complete after 1s
aws_internet_gateway.igw: Destroying... [id=igw-01db69ef2b58b81eb]
aws_instance.private_instance: Still destroying... [id=i-0ff4a80b2ee4f014d, 10s elapsed]
aws_instance.public_instance: Still destroying... [id=i-0cc6be55e828e5b44, 10s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-01db69ef2b58b81eb, 10s elapsed]
aws_instance.private_instance: Still destroying... [id=i-0ff4a80b2ee4f014d, 20s elapsed]
aws_instance.public_instance: Still destroying... [id=i-0cc6be55e828e5b44, 20s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-01db69ef2b58b81eb, 20s elapsed]
aws_instance.private_instance: Still destroying... [id=i-0ff4a80b2ee4f014d, 30s elapsed]
aws_instance.public_instance: Still destroying... [id=i-0cc6be55e828e5b44, 30s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-01db69ef2b58b81eb, 30s elapsed]
aws_instance.private_instance: Destruction complete after 39s
aws_security_group.private_sg: Destroying... [id=sg-008a9513b2ae1ccba]
aws_subnet.private_subnet: Destroying... [id=subnet-0fcb09486dda34d12]
aws_instance.public_instance: Still destroying... [id=i-0cc6be55e828e5b44, 40s elapsed]
aws_subnet.private_subnet: Destruction complete after 1s
aws_security_group.private_sg: Destruction complete after 2s
aws_internet_gateway.igw: Still destroying... [id=igw-01db69ef2b58b81eb, 40s elapsed]
aws_instance.public_instance: Destruction complete after 50s
aws_subnet.public_subnet: Destroying... [id=subnet-02db91fa611c345c8]
aws_security_group.public_sg: Destroying... [id=sg-07fdacb3b8f08cb1d]
aws_internet_gateway.igw: Destruction complete after 47s
aws_subnet.public_subnet: Destruction complete after 1s
aws_security_group.public_sg: Destruction complete after 1s
aws_vpc.my_vpc: Destroying... [id=vpc-043c61751b1a667cf]
aws_vpc.my_vpc: Destruction complete after 1s

Destroy complete! Resources: 10 destroyed.
dmytro@ubuntuserver:~/dan_it_homeworks/aws-vpc-exercise$
```