

Terraform_VPC_EC2

Для початку перевіримо версію встановлених раніше terraform та CLI на машині

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

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)

```
1 variable "aws_region" {
2   description = "AWS region for deployment"
3   default     = "us-east-1"
4 }
5
6 variable "vpc_cidr" {
7   description = "CIDR block for VPC"
8   default     = "10.0.0.0/16"
9 }
10
11 variable "public_subnet_cidr" {
12   description = "CIDR block for public subnet"
13   default     = "10.0.1.0/24"
14 }
15
16 variable "private_subnet_cidr" {
17   description = "CIDR block for private subnet"
18   default     = "10.0.2.0/24"
19 }
20
21 # Optional: Add an output for easier debugging
22 output "vpc_cidr" {
23   value = var.vpc_cidr
24 }
25
26 output "public_subnet_cidr" {
27   value = var.public_subnet_cidr
28 }
29
30 output "private_subnet_cidr" {
31   value = var.private_subnet_cidr
32 }
33
34 output "aws_region" {
35   value = var.aws_region
36 }
37
```

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

```
1 output "public_instance_id" {
2   value = aws_instance.public_instance.id
3 }
4
5 output "public_instance_ip" {
6   value = aws_instance.public_instance.public_ip
7 }
8
9 output "private_instance_id" {
10  value = aws_instance.private_instance.id
11 }
12
13 output "private_instance_ip" {
14   value = aws_instance.private_instance.private_ip
15 }
16
```

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

```
1 provider "aws" {
2   region = var.aws_region
3 }
4
5 resource "aws_vpc" "my_vpc" {
6   cidr_block      = var.vpc_cidr
7   enable_dns_support = true
8   enable_dns_hostnames = true
9   tags = {
10     Name = "MyVPC"
11   }
12 }
13
14 resource "aws_subnet" "public_subnet" {
15   vpc_id            = aws_vpc.my_vpc.id
16   cidr_block        = var.public_subnet_cidr
17   map_public_ip_on_launch = true
18   tags = {
19     Name = "PublicSubnet"
20   }
21 }
22
23 resource "aws_subnet" "private_subnet" {
24   vpc_id            = aws_vpc.my_vpc.id
25   cidr_block        = var.private_subnet_cidr
26   tags = {
27     Name = "PrivateSubnet"
28   }
29 }
30
31 resource "aws_internet_gateway" "igw" {
32   vpc_id = aws_vpc.my_vpc.id
33   tags = {
34     Name = "InternetGateway"
35   }
36 }
37
38 resource "aws_route_table" "public_rt" {
39   vpc_id = aws_vpc.my_vpc.id
40   route {
41     cidr_block = "0.0.0.0/0"
42     gateway_id = aws_internet_gateway.igw.id
43   }
44   tags = {
45     Name = "PublicRouteTable"
46   }
47 }
48
49 resource "aws_route_table_association" "public_rt_association" {
50   subnet_id      = aws_subnet.public_subnet.id
51   route_table_id = aws_route_table.public_rt.id
52 }
53
54 resource "aws_security_group" "public_sg" {
55   vpc_id = aws_vpc.my_vpc.id
56
57   ingress {
58     from_port = 22
59     to_port   = 22
60     protocol  = "tcp"
61     cidr_blocks = ["0.0.0.0/0"]
62   }
63
64   ingress {
65     from_port = 80
66     to_port   = 80
67     protocol  = "tcp"
68     cidr_blocks = ["0.0.0.0/0"]
69   }
70
71   egress {
72     from_port = 0
73     to_port   = 0
74     protocol  = "-1"
75     cidr_blocks = ["0.0.0.0/0"]
76   }
77
78   tags = {
79     Name = "PublicSG"
80   }
81 }
82
83 resource "aws_security_group" "private_sg" {
84   vpc_id = aws_vpc.my_vpc.id
85
86   ingress {
87     from_port = 22
88     to_port   = 22
89     protocol  = "tcp"
90     security_groups = [aws_security_group.public_sg.id]
91   }
92
93   egress {
94     from_port = 0
95     to_port   = 0
96     protocol  = "-1"
97     cidr_blocks = ["0.0.0.0/0"]
98   }
99
100   tags = {
101     Name = "PrivateSG"
102   }
103 }
104
```

```

105 resource "aws_instance" "public_instance" {
106     ami           = "ami-0ed6534c7d6a8e78f"
107     instance_type = "t2.micro"
108     subnet_id     = aws_subnet.public_subnet.id
109     vpc_security_group_ids = [aws_security_group.public_sg.id]
110     associate_public_ip_address = true # Додано для публічної IP-адреси
111
112     tags = {
113         Name = "PublicInstance"
114     }
115 }
116
117 resource "aws_instance" "private_instance" {
118     ami           = "ami-0ed6534c7d6a8e78f"
119     instance_type = "t2.micro"
120     subnet_id     = aws_subnet.private_subnet.id
121     vpc_security_group_ids = [aws_security_group.private_sg.id]
122
123     tags = {
124         Name = "PrivateInstance"
125     }
126 }
127

```

Ініціалізація Terraform

```

dmytro@ubuntu-server:~/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.private_instance: Still creating... [10s elapsed]
aws_instance.private_instance: Still creating... [10s elapsed]
aws_instance.private_instance: Still creating... [20s elapsed]
aws_instance.private_instance: Still creating... [20s elapsed]
aws_instance.private_instance: Still creating... [30s elapsed]
aws_instance.private_instance: Still creating... [30s elapsed]
aws_instance.private_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 1m0s [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@ubuntu-server:~/dan_it_homeworks/aws-vpc-exercise$

```

Перевірка інстансів

Instances (2) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	PublicInstance	i-0cc6be55e828e5b44	Running	t2.micro	Initializing	View alarms +	us-east-1e	ec2-54-172-103-39.co...	54.172.103.39	-
<input type="checkbox"/>	PrivateInstance	i-0ff4a80b2ee4f014d	Running	t2.micro	Initializing	View alarms +	us-east-1e	-	-	-

Public IPv4 ...

Elastic IP

IPv6 IPs

Monitoring

Security group name

Key name

Launch time

Platfor...

54.172.103.39	-	-	disabled	terraform-2024111411...	-	2024/11/14 14:58 GMT+2	Linux/UNIX
-	-	-	disabled	terraform-2024111411...	-	2024/11/14 14:58 GMT+2	Linux/UNIX

Instances (1/2) Info

Last updated 3 minutes ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input checked="" type="checkbox"/>	PublicInstance	i-0cc6be55e828e5b44	Running	t2.micro	Initializing	View alarms +	us-east-1e	ec2-54-172-103-39.co...	54.172.103.39	-
<input type="checkbox"/>	PrivateInstance	i-0ff4a80b2ee4f014d	Running	t2.micro	Initializing	View alarms +	us-east-1e	-	-	-

i-0cc6be55e828e5b44 (PublicInstance)

Instance ID

i-0cc6be55e828e5b44

IPv6 address

-

Hostname type

IP name: ip-10-0-1-187.ec2.internal

Answer private resource DNS name

-

Auto-assigned IP address

54.172.103.39 [Public IP]

IAM Role

-

IMDSv2

Optional

Public IPv4 address

54.172.103.39 | open address

Instance state

Running

Private IP DNS name (IPv4 only)

ip-10-0-1-187.ec2.internal

Instance type

t2.micro

VPC ID

vpc-043c61751b1a667cf (MyVPC)

Subnet ID

subnet-02db91fa611c345c8 (PublicSubnet)

Instance ARN

arn:aws:ec2:us-east-1:746669199028:instance/i-0cc6be55e828e5b44

Private IPv4 addresses

10.0.1.187

Public IPv4 DNS

ec2-54-172-103-39.compute-1.amazonaws.com | open address

Elastic IP addresses

-

AWS Compute Optimizer finding

User: am:aws:iam::746669199028:user(dimonchik06@gmail.com) is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action

Retry

Auto Scaling Group name

-

Instances (1/2) Info

Last updated 1 minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	PublicInstance	i-0cc6be55e828e5b44	Running	t2.micro	Initializing	View alarms +	us-east-1e	ec2-54-172-103-39.co...	54.172.103.39	-
<input checked="" type="checkbox"/>	PrivateInstance	i-0ff4a80b2ee4f014d	Running	t2.micro	Initializing	View alarms +	us-east-1e	-	-	-

i-0ff4a80b2ee4f014d (PrivateInstance)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

Networking details Info

Public IPv4 address

-

Public IPv4 DNS

-

Subnet ID

subnet-0fcb09486dda34d12 (PrivateSubnet)

Availability zone

us-east-1e

Use RBN as guest OS hostname

Disabled

Private IPv4 addresses

10.0.2.113

Private IP DNS name (IPv4 only)

ip-10-0-2-113.ec2.internal

IPv6 addresses

-

Carrier IP addresses (ephemeral)

-

Answer RBN DNS hostname IPv4

Disabled

VPC ID

vpc-043c61751b1a667cf (MyVPC)

Secondary private IPv4 addresses

-

Outpost ID

-

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@ubuntu-server:~/dan_it_homeworks/aws-vpc-exercise$
```