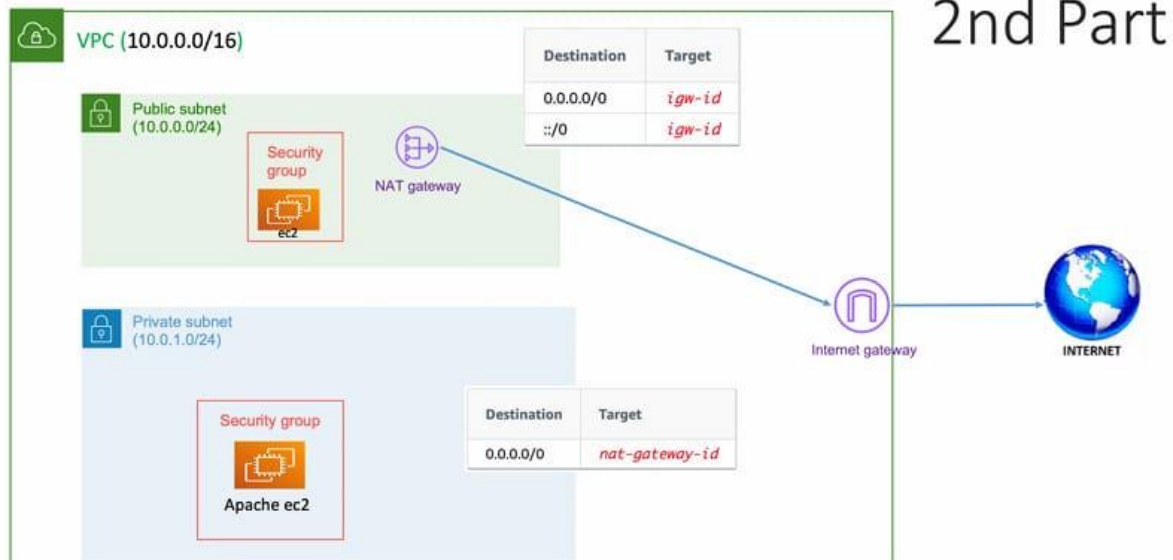


Bola Wisa

## Task 2

### Lab2:

Use variables for all arguments, output public\_ip , private\_ip , screenshot of the public ip from the browser, private ip logs also use count for creating subnets.



Search

<input type="checkbox"/>	Name	VPC ID	State	Block Public...	IPv4
<input type="checkbox"/>	VPC-Sohag	<a href="#">vpc-04a57d1693c502559</a>	Available	Off	10.0

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 D
<input checked="" type="checkbox"/>	Apache-Server	i-04cd299c7fc675abb	Running	t2.micro	...	<a href="#">View alarms +</a>	us-east-1d	-
<input type="checkbox"/>	WebServer	i-0dbad51c4740cb6d0	Running	t2.micro	...	<a href="#">View alarms +</a>	us-east-1e	ec2-100-25-15
<input checked="" type="checkbox"/>	Jump-Box	i-04625ed17a47de590	Running	t2.micro	...	<a href="#">View alarms +</a>	us-east-1e	-

> .terraform

✓ task

> .terraform

≡ .terraform.lock.hcl

Y instances.tf

Y network.tf

Y outputs.tf

Y providers.tf

Y security.tf

{ } terraform.tfstate

≡ terraform.tfstate.ba...

Y variables.tf

≡ .terraform.lock.hcl

Y instances.tf ×

terraform > task > Y instances.tf

```
1 resource "aws_instance" "jump_box" {
2     ami           = var.ami_id
3     instance_type = var.instance_type
4     subnet_id     = aws_subnet.public_subnet.id
5     key_name       = var.key_name
6     vpc_security_group_ids = [aws_security_group.web_sg.id]
7
8     tags = {
9         Name = "Jump-Box"
10    }
11 }
12
13 resource "aws_instance" "apache_server" {
14     ami           = var.ami_id
15     instance_type = var.instance_type
16     subnet_id     = aws_subnet.private_subnet.id
17     key_name       = var.key_name
18     vpc_security_group_ids = [aws_security_group.private_sg.id]
19
20     user_data = <<-EOF
21     |         |         |         |         |
22     |         |         |         |         | #!/bin/bash
23     |         |         |         |         | sudo dnf update -y
24     |         |         |         |         | sudo dnf install -y httpd
25     |         |         |         |         | sudo systemctl start httpd
26     |         |         |         |         | sudo systemctl enable httpd
27     |         |         |         |         | echo "<h1>Welcome Bola, Today is Wednesday 26-3-2025</h1>" | sudo tee /var/www/html/index.html
28     |         |         |         |         | EOF
29
30     tags = {
31         Name = "Apache-Server"
32     }
33 }
34 }
```

terraform &gt; task &gt; network.tf

```
1 resource "aws_vpc" "main" {
2   cidr_block = var.vpc_cidr
3
4   tags = {
5     Name = "VPC-Sohag"
6   }
7 }
8
9 resource "aws_subnet" "public_subnet" {
10  vpc_id          = aws_vpc.main.id
11  cidr_block      = var.subnet_cidrs[0]
12  map_public_ip_on_launch = true
13
14  tags = {
15    Name = "Public-Subnet"
16  }
17 }
18
19 resource "aws_subnet" "private_subnet" {
20  vpc_id          = aws_vpc.main.id
21  cidr_block      = var.subnet_cidrs[1]
22
23  tags = {
24    Name = "Private-Subnet"
25  }
26 }
27
28 resource "aws_internet_gateway" "igw" {
29  vpc_id = aws_vpc.main.id
30
31  tags = {
32    Name = "Internet-Gateway"
33  }
34 }
35
36 resource "aws_route_table" "public_rt" {
37  vpc_id = aws_vpc.main.id
38
39  route {
40    cidr_block = "0.0.0.0/0"
41    gateway_id = aws_internet_gateway.igw.id
42  }
43
44  tags = {
45    Name = "Public-RT"
```

terraform > task > network.tf

```
36 resource "aws_route_table" "public_rt" {
44     tags = {
45         Name = "Public-RT"
46     }
47 }
48
49 resource "aws_route_table_association" "public_assoc" {
50     subnet_id      = aws_subnet.public_subnet.id
51     route_table_id = aws_route_table.public_rt.id
52 }
53
54 resource "aws_eip" "nat_eip" {
55     domain = "vpc"
56 }
57
58 resource "aws_nat_gateway" "nat" {
59     allocation_id = aws_eip.nat_eip.id
60     subnet_id     = aws_subnet.public_subnet.id
61
62     tags = {
63         Name = "NAT-Gateway"
64     }
65 }
66
67 resource "aws_route_table" "private_rt" {
68     vpc_id = aws_vpc.main.id
69
70     route {
71         cidr_block      = "0.0.0.0/0"
72         nat_gateway_id = aws_nat_gateway.nat.id
73     }
74
75     tags = {
76         Name = "Private-RT"
77     }
78 }
79
80 resource "aws_route_table_association" "private_assoc" {
81     subnet_id      = aws_subnet.private_subnet.id
82     route_table_id = aws_route_table.private_rt.id
83 }
84
```

outputs.tf ✕

terraform > task > outputs.tf

```
1 output "jump_box_public_ip" {
2     description = "Public IP of the Jump Box"
3     value       = aws_instance.jump_box.public_ip
4 }
5
6 output "apache_server_private_ip" {
7     description = "Private IP of the Apache Server"
8     value       = aws_instance.apache_server.private_ip
9 }
10
```

terraform > task > providers.tf

```
1 provider "aws" {
2     region = var.aws_region
3 }
4
```

terraform > task > security.tf

```
1 # Security Group for Public Instance (Web-SG)
2 resource "aws_security_group" "web_sg" {
3     vpc_id = aws_vpc.main.id
4     name   = "Web-SG"
5
6     tags = {
7         Name = "Web-SG"
8     }
9 }
10
11 # Allow HTTP from anywhere
12 resource "aws_vpc_security_group_ingress_rule" "allow_http" {
13     security_group_id = aws_security_group.web_sg.id
14     cidr_ipv4         = "0.0.0.0/0"
15     from_port         = 80
16     to_port           = 80
17     ip_protocol       = "tcp"
18 }
19
20 # Allow SSH from anywhere (For public instance)
21 resource "aws_vpc_security_group_ingress_rule" "allow_ssh_public" {
22     security_group_id = aws_security_group.web_sg.id
23     cidr_ipv4         = "0.0.0.0/0"
24     from_port         = 22
25     to_port           = 22
26     ip_protocol       = "tcp"
27 }
28
29 # Allow all outbound traffic (for Web-SG)
30 resource "aws_vpc_security_group_egress_rule" "allow_all_out_web" {
```

terraform > task > security.tf

```
30 resource "aws_vpc_security_group_egress_rule" "allow_all_out_web" {
31     cidr_ipv4         = "0.0.0.0/0"
32     ip_protocol       = "-1" # All protocols
33 }
34
35
36 # Security Group for Private Instance (Private-SG)
37 resource "aws_security_group" "private_sg" {
38     vpc_id = aws_vpc.main.id
39     name   = "Private-SG"
40
41     tags = {
42         Name = "Private-SG"
43     }
44 }
45
46 # Allow SSH only from the Public Subnet (Public Instance)
47 resource "aws_vpc_security_group_ingress_rule" "allow_ssh_private" {
48     security_group_id = aws_security_group.private_sg.id
49     referenced_security_group_id = aws_security_group.web_sg.id
50     from_port         = 22
51     to_port           = 22
52     ip_protocol       = "tcp"
53 }
54
55 # Allow all outbound traffic (for Private-SG)
56 resource "aws_vpc_security_group_egress_rule" "allow_all_out_private" {
57     security_group_id = aws_security_group.private_sg.id
58     cidr_ipv4         = "0.0.0.0/0"
59     ip_protocol       = "-1" # All protocols
60 }
```

security.tf

variables.tf X

terraform &gt; task &gt; variables.tf

```
1 variable "aws_region" {
2     description = "AWS region to deploy resources"
3     type        = string
4     default     = "us-east-1"
5 }
6
7 variable "vpc_cidr" {
8     description = "CIDR block for the VPC"
9     type        = string
10    default     = "10.0.0.0/16"
11 }
12
13 variable "subnet_cidrs" {
14     description = "List of CIDR blocks for public and private subnets"
15     type        = list(string)
16     default     = ["10.0.0.0/24", "10.0.1.0/24"]
17 }
18
19 variable "key_name" {
20     description = "The name of the SSH key pair"
21     type        = string
22     default     = "vockey"
23 }
24
25 variable "instance_type" {
26     description = "EC2 instance type"
27     type        = string
28     default     = "t2.micro"
29 }
30
31 variable "ami_id" {
32     description = "AMI ID for the instances"
33     type        = string
34     default     = "ami-08b5b3a93ed654d19"
35 }
36
```

Session Servers Tools Games Sessions view Split MultiExec Tunneling Packages Settings Help

Quick connect... 16. 23.23.72.217 (ec2-user) X +

/home/ec2-user/

Name

- ..
- .ssh
- labsuser.pem
- .bashrc
- .bash\_profile
- .bash\_logout
- .bash\_history

```
[ec2-user@ip-10-0-0-250 ~]$ curl http://10.0.1.120
<h1>Welcome Bola, Today is Wednesday 26-3-2025</h1>
[ec2-user@ip-10-0-0-250 ~]$ curl -v http://10.0.1.120
* Trying 10.0.1.120:80...
* Connected to 10.0.1.120 (10.0.1.120) port 80
> GET / HTTP/1.1
> Host: 10.0.1.120
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Wed, 26 Mar 2025 17:57:42 GMT
< Server: Apache/2.4.62 (Amazon Linux)
< Last-Modified: Wed, 26 Mar 2025 16:37:27 GMT
< ETag: "34-6314173e0dcc9"
< Accept-Ranges: bytes
< Content-Length: 52
< Content-Type: text/html; charset=UTF-8
<
<h1>Welcome Bola, Today is Wednesday 26-3-2025</h1>
* Connection #0 to host 10.0.1.120 left intact
[ec2-user@ip-10-0-0-250 ~]$
```

