

Terraform_Task_2

Use variables for all arguments, output public_ip , private_ip , screenshot of the public ip from the browser, private ip logs also use count or for each for creating subnets

The Solution

First The Steps Should Be Like This

- 1 - Create VPC and use Sprints as the name and 10.0.0.0/16 as the CIDR block then click create vpc.
- 2 - use for each to create 2 subnet on this vpc (1 public subnet with "Public_Sub_Sprints" as name and "10.0.0.0/24" as the CIDR block) (1 private subnet with r "Private_Sub_Sprints" as the name, and "10.0.1.0/24" as the CIDR block)
- 3 - Create Internet Gateway and use "Sprints-IGW" as the name and attach it for this vpc
- 4 - Create NAT Gateway And use "Sprints-NAT " as the name and Choose Sprints-IGW as the gateway with Public_Sub_Sprints as the subnet\
- 5 - Create Route Table And use "Public-RT" as the name, select Sprints VPC and destination 0.0.0.0/0 and target Sprints-IGW as the gateway
- 6 - Create Route Table And use "Private-RT" as the name, select Sprints VPC and destination 0.0.0.0/0 and target Sprints-NAT gateway
- 7 - Create 1 Security Group with "Security_Sprints_Public" as name and allow HTTP and HTTPS and ssh traffic
- 8 - Search for the Ubuntu AMI ID.
- 9 - use for each for Create 2 EC2 instance (1 instance for public subnet and install nginx on it with enable public ip) and (1 instance for private subnet and install apache on it with disable public ip)
- 10 - generate output logs of both instance

The Final Code is:

Variables.tf

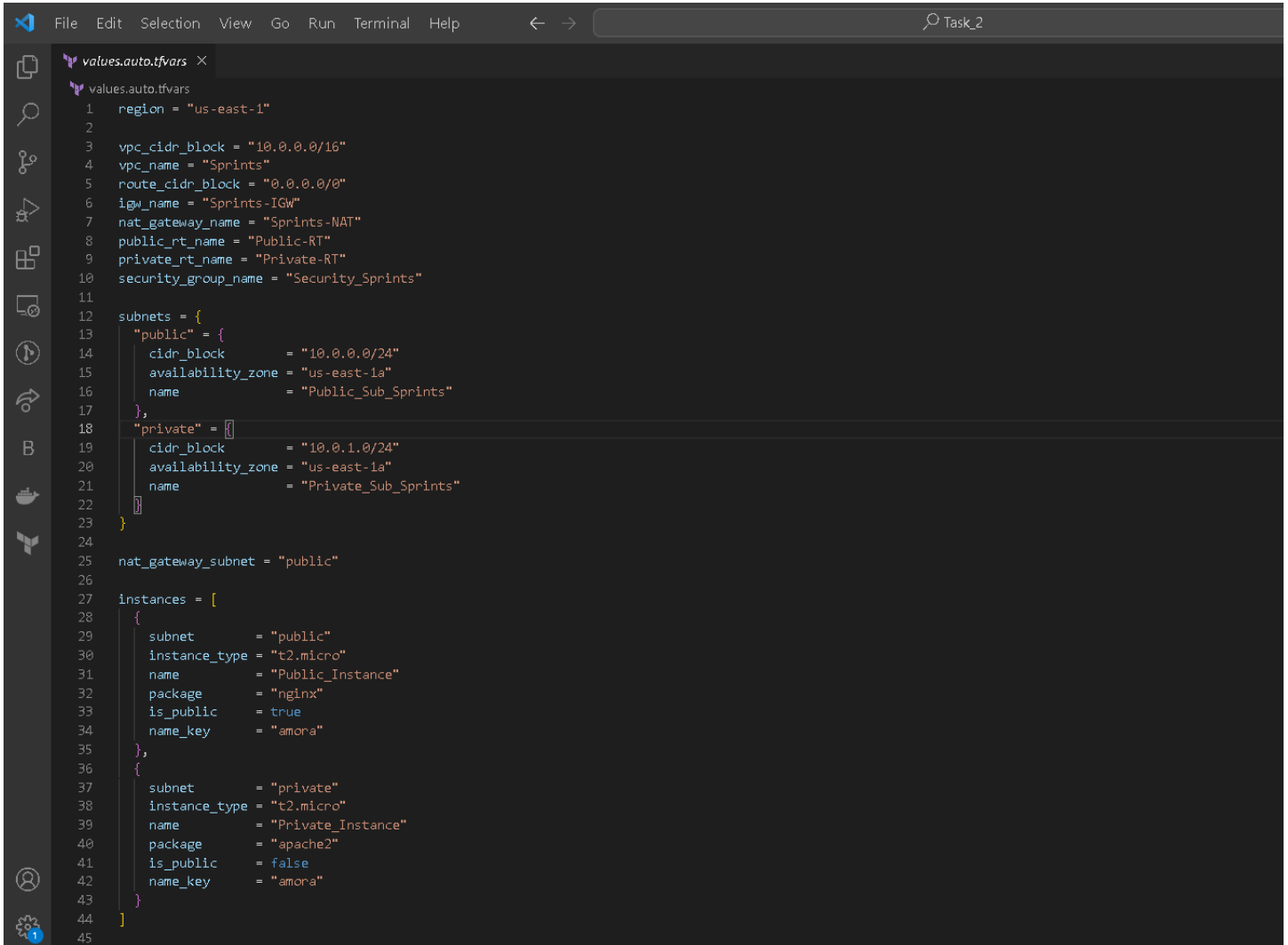
```
variables.tf
1  variable "region" {
2    | description = "AWS region"
3  }
4
5  variable "vpc_cidr_block" {
6    | description = "CIDR block for the VPC"
7  }
8
9  variable "route_cidr_block" {
10   | description = "CIDR block for the Route"
11 }
12
13 variable "vpc_name" {
14   | description = "Name of the VPC"
15 }
16
17 variable "igw_name" {
18   | description = "Name of the Internet Gateway"
19 }
20
21 variable "nat_gateway_name" {
22   | description = "Name of the NAT Gateway"
23 }
24
25 variable "public_rt_name" {
26   | description = "Name of the public route table"
27 }
28
29 variable "private_rt_name" {
30   | description = "Name of the private route table"
31 }
32
33 variable "security_group_name" {
34   | description = "Name of the security group"
35 }
36
37 variable "subnets" {
38   | description = "Configuration for subnets"
39
40   type = map(object({
41     cidr_block      = string
42     availability_zone = string
43     name            = string
44   }))
45 }
```

variables.tf X

variables.tf

```
34 | description = "Name of the security group"
35 | }
36 |
37 | variable "subnets" {
38 |   description = "Configuration for subnets"
39 |
40 |   type = map(object({
41 |     cidr_block      = string
42 |     availability_zone = string
43 |     name            = string
44 |   }))
45 | }
46 |
47 | variable "nat_gateway_subnet" {
48 |   description = "Subnet index for NAT Gateway"
49 | }
50 |
51 | variable "instances" {
52 |   description = "Configuration for instances"
53 |
54 |   type = list(object({
55 |     subnet          = string
56 |     instance_type    = string
57 |     name             = string
58 |     package          = string
59 |     is_public        = bool
60 |     name_key         = string
61 |   }))
62 | }
63 |
64 |
```

values.auto.tfvars



```
1  region = "us-east-1"
2
3  vpc_cidr_block = "10.0.0.0/16"
4  vpc_name = "Sprints"
5  route_cidr_block = "0.0.0.0/0"
6  igw_name = "Sprints-IGW"
7  nat_gateway_name = "Sprints-NAT"
8  public_rt_name = "Public-RT"
9  private_rt_name = "Private-RT"
10 security_group_name = "Security_Sprints"
11
12 subnets = {
13   "public" = {
14     cidr_block      = "10.0.0.0/24"
15     availability_zone = "us-east-1a"
16     name            = "Public_Sub_Sprints"
17   },
18   "private" = {
19     cidr_block      = "10.0.1.0/24"
20     availability_zone = "us-east-1a"
21     name            = "Private_Sub_Sprints"
22   }
23 }
24
25 nat_gateway_subnet = "public"
26
27 instances = [
28   {
29     subnet      = "public"
30     instance_type = "t2.micro"
31     name        = "Public_Instance"
32     package     = "nginx"
33     is_public   = true
34     name_key    = "amora"
35   },
36   {
37     subnet      = "private"
38     instance_type = "t2.micro"
39     name        = "Private_Instance"
40     package     = "apache2"
41     is_public   = false
42     name_key    = "amora"
43   }
44 ]
45
```

Main.tf

```
File Edit Selection View Go Run Terminal Help Task_2

main.tf x
1 # VPC resource
2 resource "aws_vpc" "sprints_vpc" {
3   cidr_block = var.vpc_cidr_block
4   tags = {
5     Name = var.vpc_name
6   }
7 }
8
9 # Subnet resource
10 resource "aws_subnet" "subnets" {
11   for_each = var.subnets
12
13   vpc_id      = aws_vpc.sprints_vpc.id
14   cidr_block  = each.value.cidr_block
15   availability_zone = each.value.availability_zone
16
17   tags = {
18     Name = each.value.name
19   }
20 }
21
22 # Internet Gateway resource
23 resource "aws_internet_gateway" "sprints_igw" {
24   vpc_id = aws_vpc.sprints_vpc.id
25
26   tags = {
27     Name = var.igw_name
28   }
29 }
30
31 # Elastic IP resource
32 resource "aws_eip" "sprints_eip" {
33   domain = "vpc"
34 }
35
36 # NAT Gateway resource
37 resource "aws_nat_gateway" "sprints_nat" {
38   allocation_id = aws_eip.sprints_eip.id
39   subnet_id     = aws_subnet.subnets[var.nat_gateway_subnet].id
40
41   tags = {
42     Name = var.nat_gateway_name
43   }
44 }
```

File Edit Selection View Go Run Terminal Help ← → Task_2

main.tf ×
main.tf

```
45 # Route Table for Public Subnet
46 resource "aws_route_table" "public_route_table" {
47     vpc_id = aws_vpc.sprints_vpc.id
48
49     dynamic "route" {
50         for_each = [var.route_cidr_block]
51
52         content {
53             cidr_block = route.value
54             gateway_id = aws_internet_gateway.sprints_igw.id
55         }
56     }
57
58     tags = {
59         Name = var.public_rt_name
60     }
61 }
62
63 # Route Table for Private Subnet
64 resource "aws_route_table" "private_route_table" {
65     vpc_id = aws_vpc.sprints_vpc.id
66
67     dynamic "route" {
68         for_each = [var.route_cidr_block]
69
70         content {
71             cidr_block = route.value
72             gateway_id = aws_nat_gateway.sprints_nat.id
73         }
74     }
75
76     tags = {
77         Name = var.private_rt_name
78     }
79 }
80
81 # Route Table Association for Public Subnet
82 resource "aws_route_table_association" "public_subnet_association" {
83     subnet_id      = aws_subnet.subnets["public"].id
84     route_table_id = aws_route_table.public_route_table.id
85 }
86
87 # Route Table Association for Private Subnet
88 resource "aws_route_table_association" "private_subnet_association" {
89     subnet_id      = aws_subnet.subnets["private"].id
```

```
88 resource "aws_route_table_association" "private_subnet_association" {
89     subnet_id      = aws_subnet.subnets["private"].id
90     route_table_id = aws_route_table.private_route_table.id
91 }
92
93
94
95 # Security Group resource
96 resource "aws_security_group" "sprints_sg" {
97     name           = var.security_group_name
98     description    = "Security group for instances in Sprints VPC"
99
100     vpc_id = aws_vpc.sprints_vpc.id
101
102     ingress {
103         from_port = 22
104         to_port   = 22
105         protocol  = "tcp"
106         cidr_blocks = [var.route_cidr_block]
107     }
108
109     ingress {
110         from_port = 80
111         to_port   = 80
112         protocol  = "tcp"
113         cidr_blocks = [var.route_cidr_block]
114     }
115
116     ingress {
117         from_port = 443
118         to_port   = 443
119         protocol  = "tcp"
120         cidr_blocks = [var.route_cidr_block]
121     }
122
123     egress {
124         from_port = 0
125         to_port   = 0
126         protocol  = "-1"
127         cidr_blocks = [var.route_cidr_block]
128     }
129 }
130 # Search for the Ubuntu AMI ID
131
```

```

132 data "aws_ami" "ubuntu" {
133     most_recent = true
134
135     filter {
136         name   = "name"
137         values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
138     }
139
140     filter {
141         name   = "virtualization-type"
142         values = ["hvm"]
143     }
144
145     filter {
146         name   = "architecture"
147         values = ["x86_64"]
148     }
149
150     owners = ["099720109477"]
151 }
152
153 # EC2 instance resource
154 resource "aws_instance" "instances" {
155     for_each = { for instance in var.instances : instance.name => instance }
156
157     subnet_id           = aws_subnet.subnets[each.value.subnet].id
158     vpc_security_group_ids = [aws_security_group.sprints_sg.id]
159     ami                 = data.aws_ami.ubuntu.id
160     instance_type       = each.value.instance_type
161     associate_public_ip_address = each.value.is_public
162     key_name             = each.value.name_key
163     user_data = <<-EOF
164     #!/bin/bash
165     sudo apt-get update
166     sudo apt-get install -y ${each.value.package}
167     sudo systemctl enable ${each.value.package}
168     sudo systemctl start ${each.value.package}
169     EOF
170
171     tags = {
172         Name = each.value.name
173     }
174
175 }

```

Outputs.tf

```

outputs.tf ×
outputs.tf
1  output "public_ip" {
2      value = [for instance in aws_instance.instances : instance.public_ip]
3  }
4
5  output "private_ip" {}
6      value = [for instance in aws_instance.instances : instance.private_ip]
7  }

```


Provider.tf

```
providers.tf X
providers.tf
1  # Provider configuration
2  provider "aws" {
3      region = var.region
4  }
5
```

The final result in terminal

```
amr@DESKTOP-D5VVHNO: ~/  X + v
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_eip.sprints_eip: Creating...
aws_vpc.sprints_vpc: Creating...
aws_eip.sprints_eip: Creation complete after 2s [id=eipalloc-0ef4e3edaf553f6a8]
aws_vpc.sprints_vpc: Creation complete after 4s [id=vpc-08b012cd3563146af]
aws_internet_gateway.sprints_igw: Creating...
aws_subnet.subnets["public"]: Creating...
aws_subnet.subnets["private"]: Creating...
aws_security_group.sprints_sg: Creating...
aws_subnet.subnets["private"]: Creation complete after 1s [id=subnet-0c34f58a6e81aa73f]
aws_internet_gateway.sprints_igw: Creation complete after 1s [id=igw-03488a0acaeb9235a]
aws_route_table.public_route_table: Creating...
aws_subnet.subnets["public"]: Creation complete after 1s [id=subnet-0183a7398d2c249be]
aws_nat_gateway.sprints_nat: Creating...
aws_route_table.public_route_table: Creation complete after 2s [id=rtb-0f5b5ff756d48e7c8]
aws_route_table_association.public_subnet_association: Creating...
aws_route_table_association.public_subnet_association: Creation complete after 1s [id=rtbassoc-0ac926f9df7ee43f1]
aws_security_group.sprints_sg: Creation complete after 4s [id=sg-087d41880cccb0f01]
aws_instance.instances["Public_Instance"]: Creating...
aws_instance.instances["Private_Instance"]: Creating...
aws_nat_gateway.sprints_nat: Still creating... [10s elapsed]
```

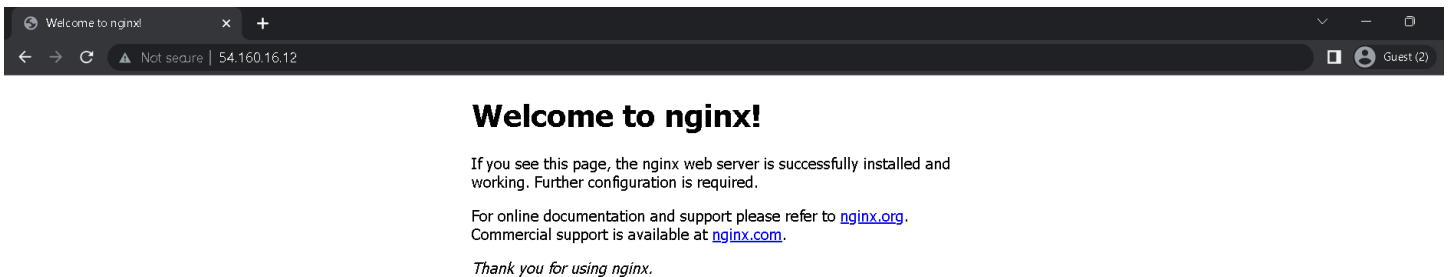
```
amr@DESKTOP-D5VVHN0: ~/ terraform_tasks/Task_2$ terraform apply
aws_nat_gateway.sprints_nat: Still creating... [40s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [50s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m0s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m10s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m20s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m30s elapsed]
aws_nat_gateway.sprints_nat: Still creating... [1m40s elapsed]
aws_nat_gateway.sprints_nat: Creation complete after 1m46s [id=nat-06b8c7498d9e027c6]
aws_route_table.private_route_table: Creating...
aws_route_table.private_route_table: Creation complete after 3s [id=rtb-0b7e97ebb4cb02292]
aws_route_table_association.private_subnet_association: Creating...
aws_route_table_association.private_subnet_association: Creation complete after 1s [id=rtbassoc-03411c08fc5c1cbc0]

Apply complete! Resources: 13 added, 0 changed, 0 destroyed.

Outputs:

private_ip = [
  "10.0.1.64",
  "10.0.0.187",
]
public_ip = [
  "",
  "54.160.16.12",
]
amr@DESKTOP-D5VVHN0: ~/terraform_tasks/Task_2$
```

The Public Ip Results In Browser



The Private Ip from logs

```
ubuntu@ip-10-0-1-64: ~  
Reporting Problems  
</div>  
<div class="content_section_text">  
  <p>  
    Please use the <tt>ubuntu-bug</tt> tool to report bugs in the  
    Apache2 package with Ubuntu. However, check <a  
    href="https://bugs.launchpad.net/ubuntu/+source/apache2" ←  
    rel="nofollow">existing bug reports</a> before reporting a new bug.  
  </p>  
  <p>  
    Please report bugs specific to modules (such as PHP and others)  
    to respective packages, not to the web server itself.  
  </p>  
</div>  
  
</div>  
</div>  
<div class="validator">  
</div>  
</body>  
</html>  
✓  
ubuntu@ip-10-0-1-64: $
```