Terraform Assignment 01 - Haneef Shaikh

Que 1 →

● Create one IAM user and one IAM Group using Terraform.

● Make sure you will use variables for names of IAM users and Group.

● Note :- Below files are required.

- main.tf

```
#AWS Provider
terraform {
 required_providers {
   aws = {
     source = "hashicorp/aws"
     version = "4.52.0"
   }
 }
}
provider "aws" {
 # Configuration options
}

// IAM GROUP

resource "aws_iam_group" "application_group" {
 name = var.iam_group_name
 path = var.iam_group_path
}

// IAM USER

resource "aws_iam_user" "application_users" {
 name = var.iam_user_name
 path = var.iam_user_path
}

// IAM GROUP MEMBER

resource "aws_iam_user_group_membership" "application_group_members" {
 user = aws_iam_user.application_users.name
 groups = [
   aws_iam_group.application_group.name
 ]
}
```

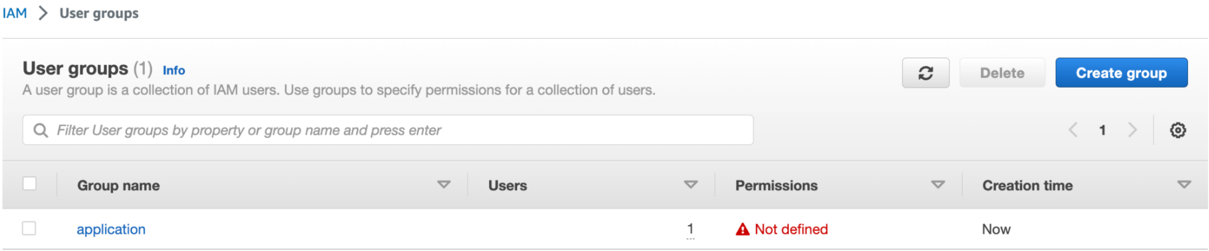
- **variables.tf**

```
// IAM GROUP

variable "iam_group_name" {
   type = string
}

variable "iam_group_path" {
   type = string
}

// IAM USER

variable "iam_user_name" {
   type = string
}

variable "iam_user_path" {
   type = string
}
```

- **terraform.tfvars**

```
// IAM GROUP
iam_group_name = "application"
iam_group_path = "/users/"

// IAM USER
iam_user_name = "apps01"
iam_user_path = "/system/"
```

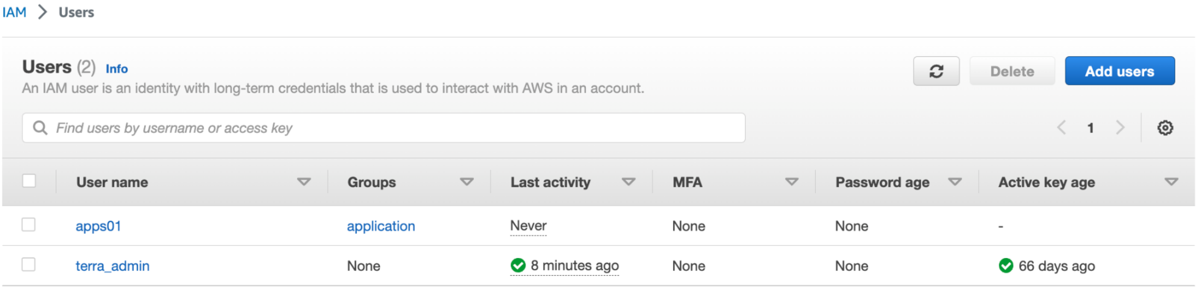
- **User and group create also assigned the created user into the created group**


IAM > User groups
Delete
Create group
Info
Filter User groups by property or group name and press enter
1

User groups (1)

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

□	Group name	Users	Permissions	Creation time
□	application	1	⚠ Not defined	Now



Users (2)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name	Groups	Last activity	MFA	Password age	Active key age
□	apps01	application	Never	None	None	-
□	terra_admin	None	✓ 8 minutes ago	None	None	✓ 66 days ago

Que 2 →

● Create one EC2 Instance and Elastic IP using Terraform and Map elastic IP with EC2 instance.

● Also please make sure you will use a combination of both variables in the main.tf file.

○ i.e. local and variable from variables.tf and custom.tfvars file.

● Also use output values to print EC2 instances Public DNS name , Private DNS name , Private IP and Public IP.

● Note :-

○ Here you will require one locals in the main.tf file.

○ Also four output values in the main.tf file.

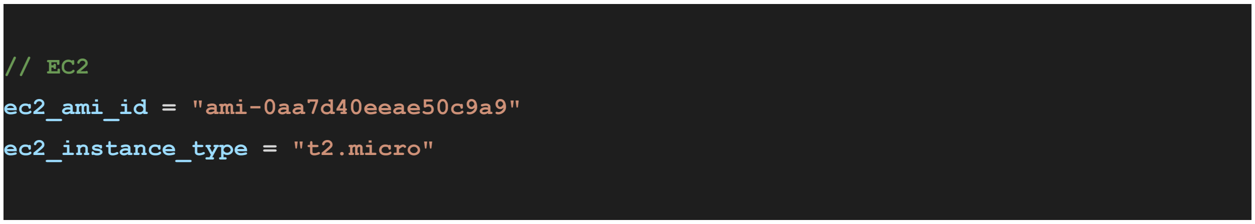
Main.tf

```
#AWS Provider
terraform {
 required_providers {
   aws = {
     source = "hashicorp/aws"
     version = "4.52.0"
   }
 }
}

provider "aws" {
 # Configuration options
}

locals {
   common_tags = {
     user = "devops"
   }
}

// EIP
resource "aws_eip" "lb" {
 vpc     = true
 tags    = local.common_tags
}

// EC2 INSTANCE
resource "aws_instance" "my_ec2" {
   ami              = var.ec2_ami_id
   instance_type = var.ec2_instance_type
   tags             = local.common_tags
}

// EIP TO EC2

resource "aws_eip_association" "myeip_assoc" {
 instance_id    = aws_instance.my_ec2.id
 allocation_id = aws_eip.lb.id
}

// OUTPUT

output "OUT_EIP_public_dns" {
 value = aws_eip.lb.public_dns
}
output "OUT_EIP_private_dns" {
 value = aws_eip.lb.private_ip
}
output "OUT_EC2_public_ip" {
 value = aws_instance.my_ec2.public_ip
}
output "OUT_EC2_private_ip" {
 value = aws_instance.my_ec2.private_ip
}
```

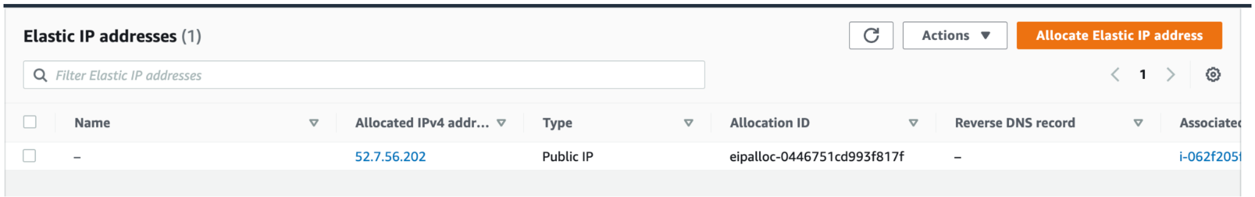
Variable.tf

```

// EC2

variable "ec2_ami_id" {
   type = string
}

variable "ec2_instance_type" {
   type = string
}
```

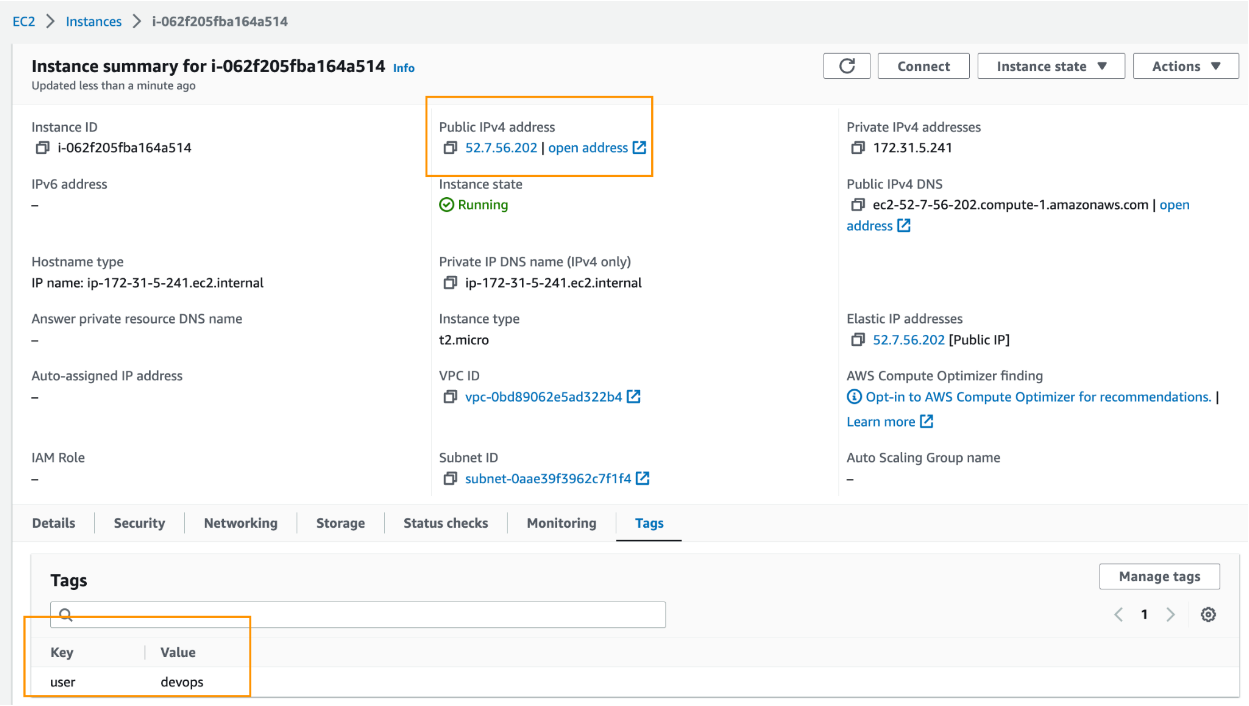
Terraform.tfvars



```
// EC2
ec2_ami_id = "ami-0aa7d40eeae50c9a9"
ec2_instance_type = "t2.micro"
```



□	Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record	Associated
□	–	52.7.56.202	Public IP	eipalloc-0446751cd993f817f	–	i-062f205



Instance summary for i-062f205fba164a514 Info

Updated less than a minute ago

Instance ID
i-062f205fba164a514

Public IPv4 address
52.7.56.202 | open address

Private IPv4 addresses
172.31.5.241

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-52-7-56-202.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-5-241.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-5-241.ec2.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
52.7.56.202 [Public IP]

Auto-assigned IP address
–

VPC ID
vpc-0bd89062e5ad322b4

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role
–

Subnet ID
subnet-0aae39f3962c7f1f4

Auto Scaling Group name
–

Tags

Key	Value
user	devops

Que 3 →

- **Create AWS VPC with Terraform.**
- **Please follow the given link for more on AWS VPC creation.**

1. Create a VPC
2. Create 2 Public Subnet & Create 2 Private Subnet
3. Create IGW (Internet Gateway) & Attach to the VPC
4. Create Public and Private Route Table
5. Add IGW in Public Route table (0.0.0.0/0)
6. Add Public Subnet (1a & 1b) in Route table
7. Create a NAT Gateway in Public Subnet
8. Add NAT GW into the Private Route Table
9. Add Private Subnet in Private Route Table

- **Link :-**
  - ***https://varunmanik1.medium.com/how-to-create-aws-vpc-in-10-steps-less-than-5-min-a49ac12064aa***
- **Note :-**
  - **Try to create VPC manually to understand the concepts and then go for Terraform automation.**

Main.tf

```
#AWS Provider

terraform {
 required_providers {
   aws = {
     source = "hashicorp/aws"
     version = "4.52.0"
   }
 }
}

provider "aws" {
 # Configuration options
}
```

Vpc.tf

```

locals {
 common_tags = {
     user = "devops"
   }
```

```
}

#VPC
resource "aws_vpc" "cloudethix-vpc" {
 cidr_block       = var.vpc_cidr_block
 instance_tenancy = "default"
 tags             = local.common_tags
}

#private subnets
resource "aws_subnet" "cloudethix-sub-private01" {
 vpc_id                  = aws_vpc.cloudethix-vpc.id
 cidr_block              = var.private_subnet_cidr[0]
 availability_zone       = var.availability_zone[0]
 map_public_ip_on_launch = true
 tags                  = local.common_tags
}

resource "aws_subnet" "cloudethix-sub-private02" {
 vpc_id                  = aws_vpc.cloudethix-vpc.id
 cidr_block              = var.private_subnet_cidr[1]
 availability_zone       = var.availability_zone[1]
 map_public_ip_on_launch = true
 tags                    = local.common_tags
}


#public subnets
resource "aws_subnet" "cloudethix-sub-public01" {
 vpc_id                  = aws_vpc.cloudethix-vpc.id
 cidr_block              = var.public_subnet_cidr[0]
 availability_zone       = var.availability_zone[0]
 map_public_ip_on_launch = true
 tags                    = local.common_tags
}

resource "aws_subnet" "cloudethix-sub-public02" {
 vpc_id                  = aws_vpc.cloudethix-vpc.id
 cidr_block              = var.public_subnet_cidr[1]
 availability_zone       = var.availability_zone[1]
 map_public_ip_on_launch = true
 tags                    = local.common_tags
```

```
}



#Elastic IP
resource "aws_eip" "cloudethix-eip" {
 vpc      = true
 tags      = local.common_tags
}



#IGW
resource "aws_internet_gateway" "cloudethix-igw" {
 vpc_id = aws_vpc.cloudethix-vpc.id
 tags   = local.common_tags
}



#Public NAT
resource "aws_nat_gateway" "cloudethix-nat" {
 allocation_id = aws_eip.cloudethix-eip.id
 subnet_id     = aws_subnet.cloudethix-sub-public01.id
 tags          = local.common_tags
}

#Route Table
resource "aws_route_table" "cloudethix-RT-public" {
 vpc_id = aws_vpc.cloudethix-vpc.id
 tags   = local.common_tags
}

resource "aws_route_table" "cloudethix-RT-private" {
 vpc_id = aws_vpc.cloudethix-vpc.id
 tags   = local.common_tags
}

#Route
resource "aws_route" "cloudethix-route-public" {
 route_table_id            = aws_route_table.cloudethix-RT-public.id
 destination_cidr_block    = var.destination_cidr_block
 gateway_id                = aws_internet_gateway.cloudethix-igw.id
}
```

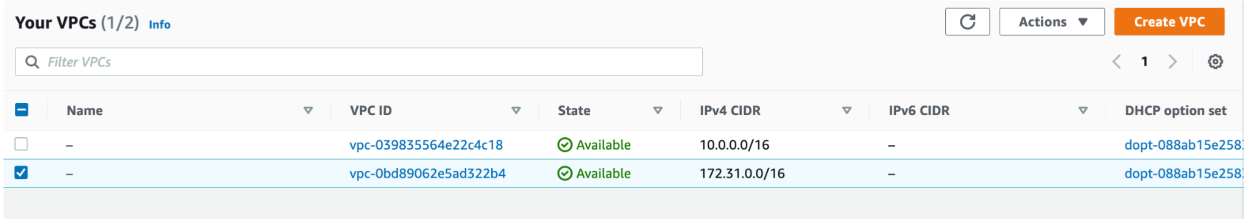
```
resource "aws_route" "cloudethix-route-private" {
 route_table_id            = aws_route_table.cloudethix-RT-private.id
 destination_cidr_block    = var.destination_cidr_block
 gateway_id                = aws_nat_gateway.cloudethix-nat.id
}

#Route Table Association
resource "aws_route_table_association" "cloudethix-RTASS-public" {
 subnet_id      = aws_subnet.cloudethix-sub-public01.id
 route_table_id = aws_route_table.cloudethix-RT-public.id
}

resource "aws_route_table_association" "cloudethix-RTASS-private" {
 subnet_id     = aws_subnet.cloudethix-sub-private01.id
 route_table_id = aws_route_table.cloudethix-RT-private.id
}
```

Variable.tf

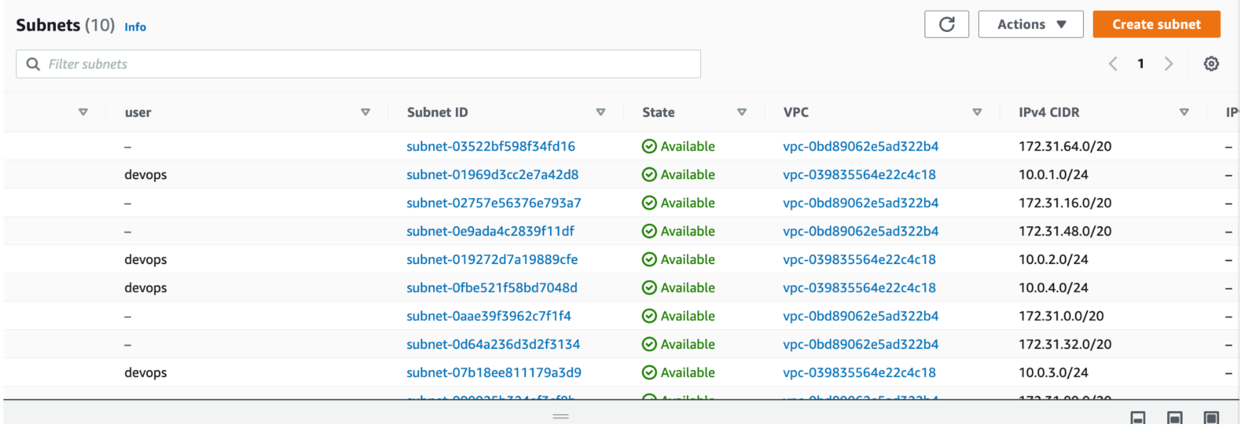
```
// VPC

variable "availability_zone" {
 type = list
}

variable "vpc_cidr_block" {
 type = string
}

variable "public_subnet_cidr" {
 type = list
}

variable "private_subnet_cidr" {
 type = list
}

variable "destination_cidr_block" {
 type = string
}
```

Terraform.tfvars

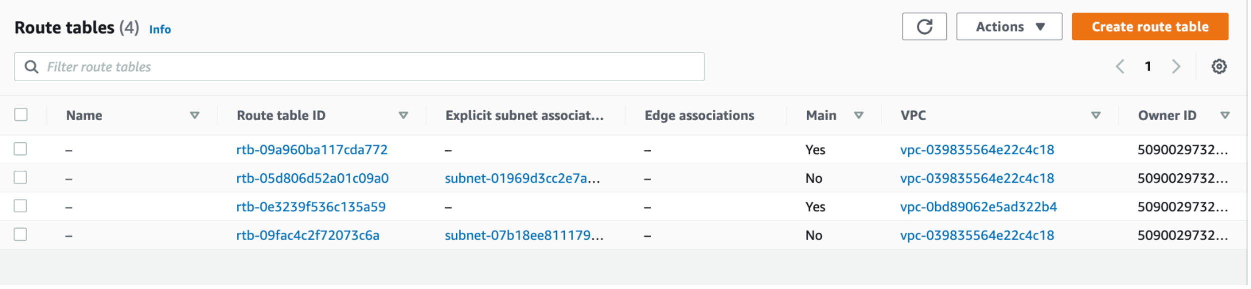
```
// VPC
availability_zone       = ["us-east-1a", "us-east-1b"]
vpc_cidr_block          = "10.0.0.0/16"
public_subnet_cidr      = ["10.0.1.0/24", "10.0.2.0/24"]
private_subnet_cidr     = ["10.0.3.0/24", "10.0.4.0/24"]
subnet_cidr             = ["10.0.1.0/24", "10.0.2.0/24", "10.0.3.0/24", "10.0.4.0/24"]
destination_cidr_block = "0.0.0.0/0"
```



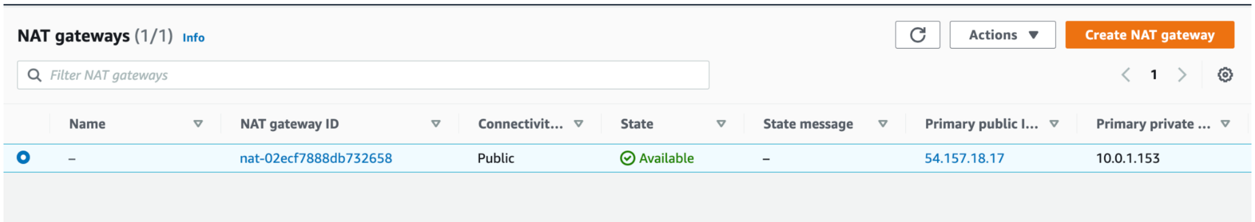
☐	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set
☐	–	vpc-039835564e22c4c18	✓ Available	10.0.0.0/16	–	dopt-088ab15e258
☑	–	vpc-0bd89062e5ad322b4	✓ Available	172.31.0.0/16	–	dopt-088ab15e258



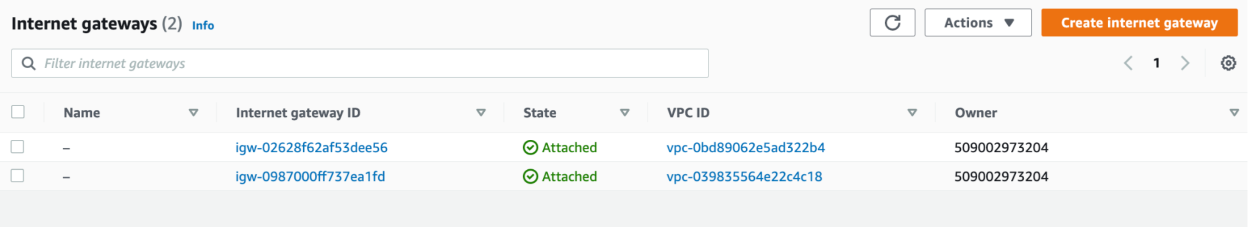
user	Subnet ID	State	VPC	IPv4 CIDR	IP
–	subnet-03522bf598f34fd16	✓ Available	vpc-0bd89062e5ad322b4	172.31.64.0/20	–
devops	subnet-01969d3cc2e7a42d8	✓ Available	vpc-039835564e22c4c18	10.0.1.0/24	–
–	subnet-02757e56376e793a7	✓ Available	vpc-0bd89062e5ad322b4	172.31.16.0/20	–
–	subnet-0e9ada4c2839f11df	✓ Available	vpc-0bd89062e5ad322b4	172.31.48.0/20	–
devops	subnet-019272d7a19889cfe	✓ Available	vpc-039835564e22c4c18	10.0.2.0/24	–
devops	subnet-0fbe521f58bd7048d	✓ Available	vpc-039835564e22c4c18	10.0.4.0/24	–
–	subnet-0aae39f3962c7f1f4	✓ Available	vpc-0bd89062e5ad322b4	172.31.0.0/20	–
–	subnet-0d64a236d3d2f3134	✓ Available	vpc-0bd89062e5ad322b4	172.31.32.0/20	–
devops	subnet-07b18ee811179a3d9	✓ Available	vpc-039835564e22c4c18	10.0.3.0/24	–



☐	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
☐	–	rtb-09a960ba117cda772	–	–	Yes	vpc-039835564e22c4c18	5090029732...
☐	–	rtb-05d806d52a01c09a0	subnet-01969d3cc2e7a...	–	No	vpc-039835564e22c4c18	5090029732...
☐	–	rtb-0e3239f536c135a59	–	–	Yes	vpc-0bd89062e5ad322b4	5090029732...
☐	–	rtb-09fac4c2f72073c6a	subnet-07b18ee811179...	–	No	vpc-039835564e22c4c18	5090029732...



	Name	NAT gateway ID	Connectivit...	State	State message	Primary public I...	Primary private ...
●	–	nat-02ecf7888db732658	Public	✓ Available	–	54.157.18.17	10.0.1.153



□	Name	Internet gateway ID	State	VPC ID	Owner
□	–	igw-02628f62af53dee56	✓ Attached	vpc-0bd89062e5ad322b4	509002973204
□	–	igw-0987000ff737ea1fd	✓ Attached	vpc-039835564e22c4c18	509002973204

Que 4 →

● Create EC2 instance one of the public Subnets of VPC that you have created & Validate your Connection using ssh.● For this You need to create below AWS resources using Terraform.

1. EC2 Instance.

2. SSH Key

3. Security Group.

● Note :-

○ Attach SSH key and Security Group to EC2 Instance using attribute reference.

○ Then try to access it from an EC2 instance using the SSH key that you have created.

Main.tf

```
#AWS Provider

terraform {
 required_providers {
   aws = {
     source = "hashicorp/aws"
     version = "4.52.0"
   }
 }
}

provider "aws" {
 # Configuration options
}
```

Ec2.tf

```

#Application EC2

 resource "aws_instance" "app" {
 ami                  = var.ec2_ami_id
 instance_type        = var.ec2_instance_type
 key_name           = aws_key_pair.cloudethix-key-pair.key_name
 security_groups    = ["${aws_security_group.cloudethix-sg-app.id}"]
 subnet_id          = aws_subnet.cloudethix-sub-public01.id
}
```

Sg.tf

```
#Application Security Group

resource "aws_security_group" "cloudethix-sg-app" {
 name        = "allow_app"
 description = "Allow app inbound traffic"
 vpc_id      = aws_vpc.cloudethix-vpc.id

 ingress {
   description      = "app from VPC"
   from_port        = 8080
   to_port          = 8080
   protocol         = "tcp"
   cidr_blocks      = ["0.0.0.0/0"]
 }
}
```

Key-pair.tf

```
#Key Pair to Access EC2
resource "aws_key_pair" "cloudethix-key-pair" {
 key_name   = "3Tier-key"
 public_key = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDkc/q0xTIzecyMPE/sjWmR9g8sP8/Xj7itL9kXRzHtYLT3T13E2OAfVC
t4zZ/eQIoTJuQWstL+slKG9anXKkrwKf4qF/2wxsZZ8Z9hUYV21KIGz9lDgmkueB3MKi07VyFhpBO1S2inbpjl
lkp0hp1AcYVOS0ulMhCC+X4y8yE5amG53/qiSLPnF0dBCa9icku0YYj6RZrjKfeL2S8uwBIMeTnPbpxn8BxkKI
djRErZjfuxASH39SYmWa7lpW3m2VReFc7t23ZjlEKFOaZWbwSK88L0EduRPV7+JbJDyCO/UxA+8E5/oJ9j9rt8
/MmE1YV5Nnf8UiHrGhH3WJkMBDZN"
}
```

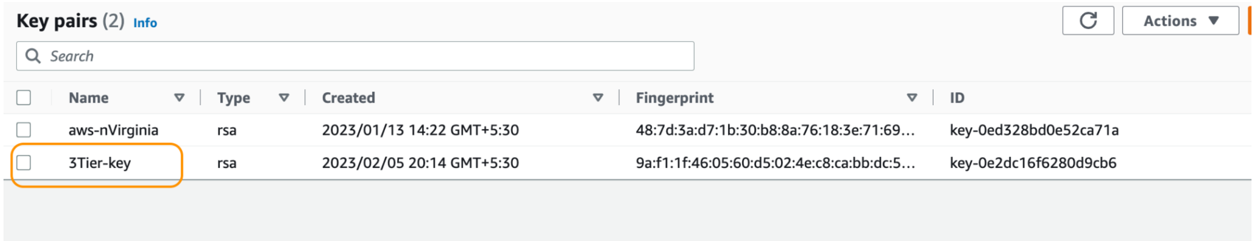
Variable.tf

```
// EC2
```

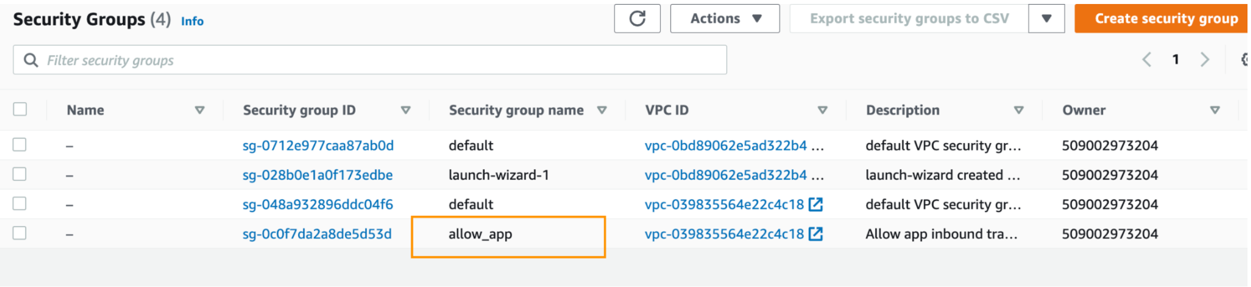
```
variable "ec2_ami_id" {
   type = string
}

variable "ec2_instance_type" {
   type = string
}
```

Terraform.tfvars

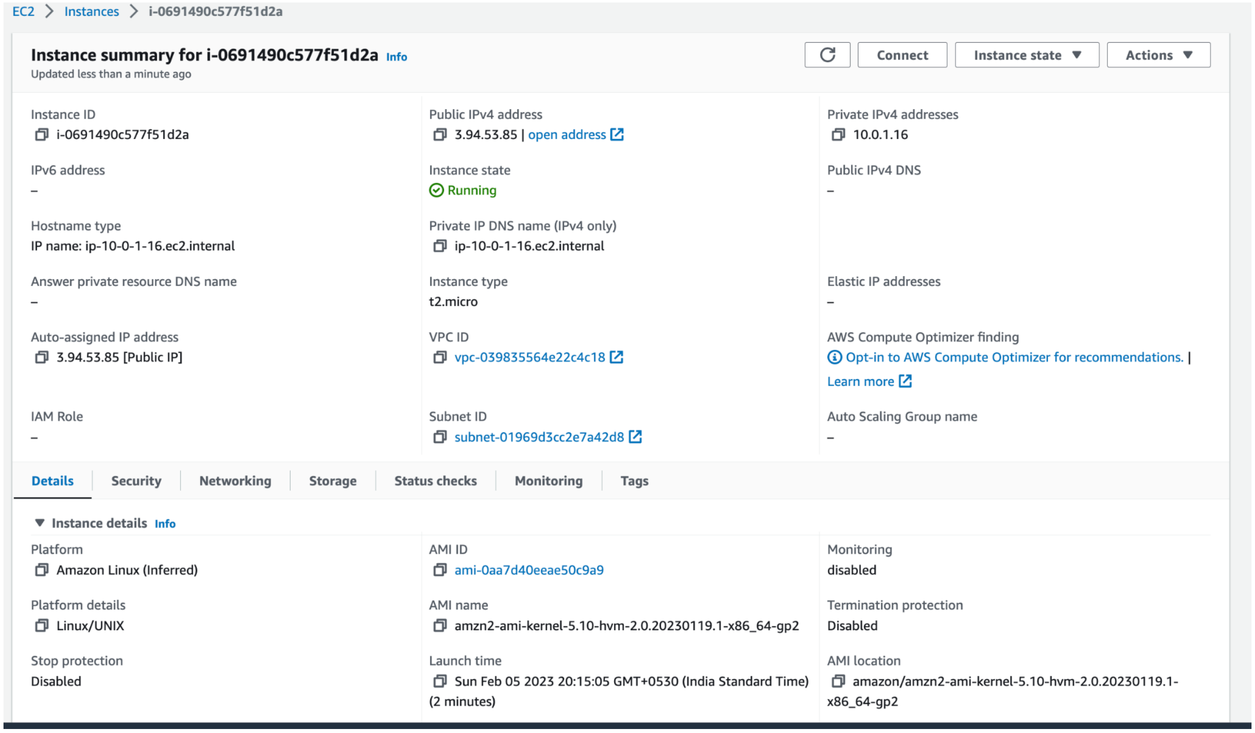
```
// EC2
ec2_ami_id           = "ami-0aa7d40eeae50c9a9"
ec2_instance_type    = "t2.micro"
```



□	Name	Type	Created	Fingerprint	ID
□	aws-nVirginia	rsa	2023/01/13 14:22 GMT+5:30	48:7d:3a:d7:1b:30:b8:8a:76:18:3e:71:69…	key-0ed328bd0e52ca71a
□	3Tier-key	rsa	2023/02/05 20:14 GMT+5:30	9a:f1:1f:46:05:60:d5:02:4e:c8:ca:bb:dc:5…	key-0e2dc16f6280d9cb6



□	Name	Security group ID	Security group name	VPC ID	Description	Owner
□	–	sg-0712e977caa87ab0d	default	vpc-0bd89062e5ad322b4 …	default VPC security gr…	509002973204
□	–	sg-028b0e1a0f173edbe	launch-wizard-1	vpc-0bd89062e5ad322b4 …	launch-wizard created …	509002973204
□	–	sg-048a932896ddc04f6	default	vpc-039835564e22c4c18	default VPC security gr…	509002973204
□	–	sg-0c0f7da2a8de5d53d	allow_app	vpc-039835564e22c4c18	Allow app inbound tra…	509002973204



Instance summary for i-0691490c577f51d2a Info

Updated less than a minute ago

Instance ID
i-0691490c577f51d2a

Public IPv4 address
3.94.53.85 | open address

Private IPv4 addresses
10.0.1.16

IPv6 address
–

Instance state
Running

Public IPv4 DNS
–

Hostname type
IP name: ip-10-0-1-16.ec2.internal

Private IP DNS name (IPv4 only)
ip-10-0-1-16.ec2.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
3.94.53.85 [Public IP]

VPC ID
vpc-039835564e22c4c18

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role
–

Subnet ID
subnet-01969d3cc2e7a42d8

Auto Scaling Group name
–

▼ Instance details Info

Platform
Amazon Linux (Inferred)

AMI ID
ami-0aa7d40eeae50c9a9

Monitoring
disabled

Platform details
Linux/UNIX

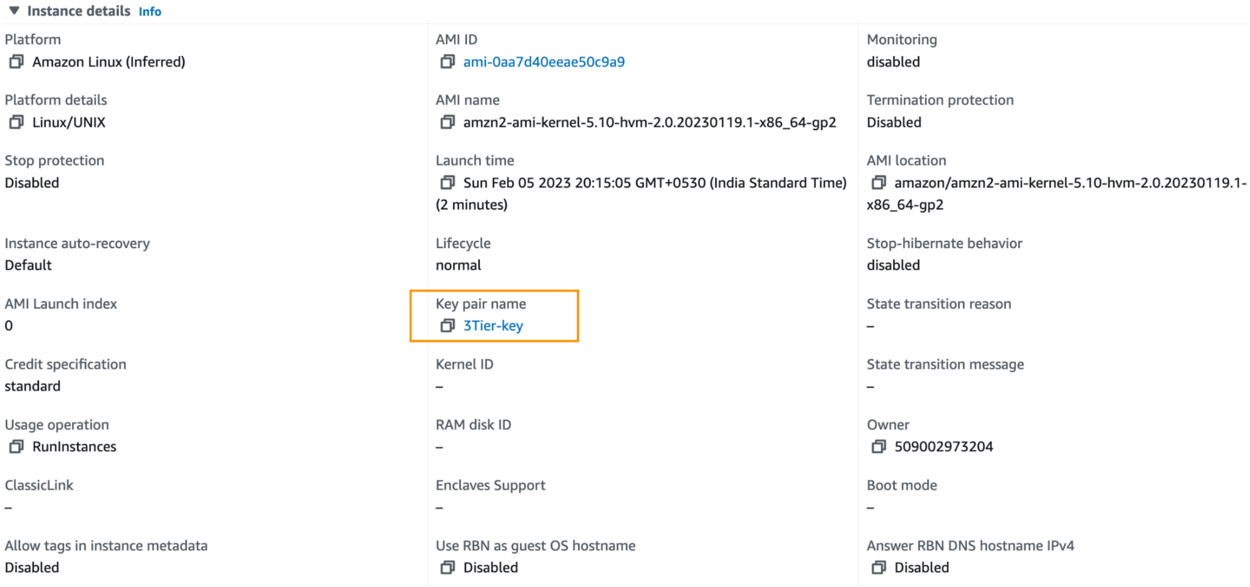
AMI name
amzn2-ami-kernel-5.10-hvm-2.0.20230119.1-x86_64-gp2

Termination protection
Disabled

Stop protection
Disabled

Launch time
Sun Feb 05 2023 20:15:05 GMT+0530 (India Standard Time) (2 minutes)

AMI location
amazon/amzn2-ami-kernel-5.10-hvm-2.0.20230119.1-x86_64-gp2



▼ Instance details Info

Platform
Amazon Linux (Inferred)

AMI ID
ami-0aa7d40eeae50c9a9

Monitoring
disabled

Platform details
Linux/UNIX

AMI name
amzn2-ami-kernel-5.10-hvm-2.0.20230119.1-x86_64-gp2

Termination protection
Disabled

Stop protection
Disabled

Launch time
Sun Feb 05 2023 20:15:05 GMT+0530 (India Standard Time) (2 minutes)

AMI location
amazon/amzn2-ami-kernel-5.10-hvm-2.0.20230119.1-x86_64-gp2

Instance auto-recovery
Default

Lifecycle
normal

Stop-hibernate behavior
disabled

AMI Launch index
0

Key pair name
3Tier-key

State transition reason
–

Credit specification
standard

Kernel ID
–

State transition message
–

Usage operation
RunInstances

RAM disk ID
–

Owner
509002973204

ClassicLink
–

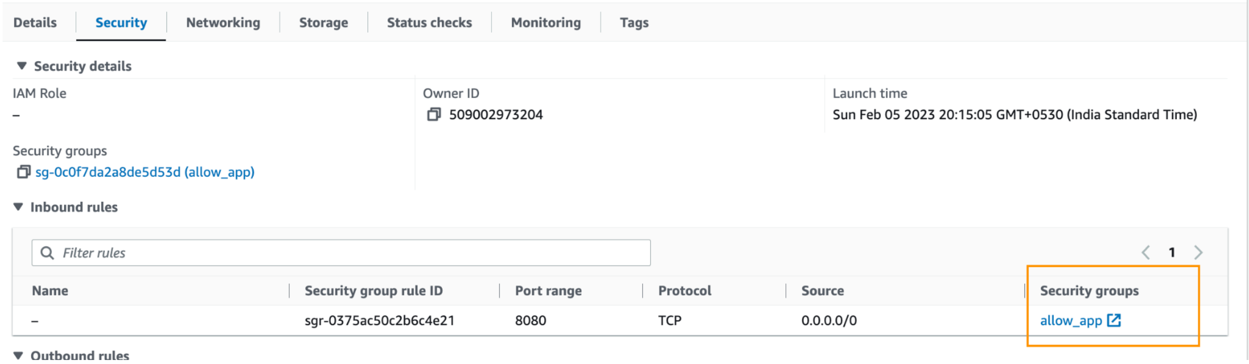
Enclaves Support
–

Boot mode
–

Allow tags in instance metadata
Disabled

Use RBN as guest OS hostname
Disabled

Answer RBN DNS hostname IPv4
Disabled



▼ Security details

IAM Role
–

Owner ID
509002973204

Launch time
Sun Feb 05 2023 20:15:05 GMT+0530 (India Standard Time)

Security groups
sg-0c0f7da2a8de5d53d (allow_app)

▼ Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
–	sgr-0375ac50c2b6c4e21	8080	TCP	0.0.0.0/0	allow_app

▼ Outbound rules