# PRAKTIKUM VPC PRIVATE DAN NAT GATEWAY

Rahadyan Danang Susetyo Pranawa – 3123522018

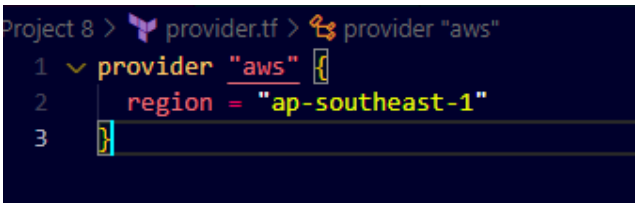
Aqil Yoga Pramono – 3123522011

R.P.A. Lexy Mangku Saputra – 312352012

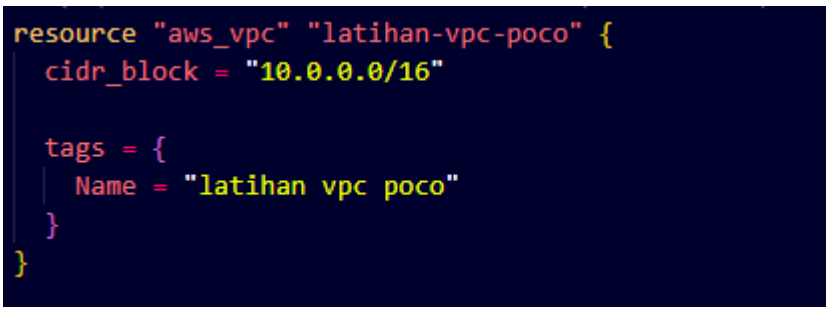
Ahmad Haidar Hafiz – 312352015

**Provider.tf**

Berisikan provider region yang kita gunakan




**Vpc.tf**




VPC (Virtual Private Cloud) adalah jaringan virtual khusus di dalam AWS yang mirip seperti jaringan lokal. Disinilah kita membuat block ip private yaitu pada cidr_block. Diberi tag Name agar lebih mudah dikenali di AWS console.


Membuat subnet public dan private, Subnet adalah bagian kecil dari jaringan VPC. Di sini, subnet dibuat dari IP 10.0.0.0 sampai. 10.0.0.255. map_public_ip_on_launch = true, EC2 yang diluncurkan di subnet ini akan otomatis mendapatkan IP publik. availability_zone = "ap-southeast-1a" di wilayah Singapura.

```
resource "aws_internet_gateway" "latihan-igw-poco" {
  vpc_id = aws_vpc.latihan-vpc-poco.id

  tags = {
    Name = "latihan igw"
  }
}

resource "aws_route_table" "latihan-public-rt-poco" {
  vpc_id = aws_vpc.latihan-vpc-poco.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.latihan-igw-poco.id
  }

  tags = {
    Name = "latihan public rt"
  }
}

resource "aws_route_table_association" "latihan-public-rta-poco" {
  subnet_id      = aws_subnet.latihan-public-subnet-poco.id
  route_table_id = aws_route_table.latihan-public-rt-poco.id
}
```

- Internet Gateway (IGW) diperlukan agar instance dalam VPC bisa mengakses dan diakses dari Internet.
- Di-attach ke VPC agar bisa dipakai oleh subnet/subnet-nya.
- Tabel routing mengatur kemana lalu lintas jaringan akan diarahkan.
- cidr_block = "0.0.0.0/0" artinya semua trafik keluar diarahkan ke Internet via Internet Gateway (gateway_id).
- Hal ini membuat subnet menjadi subnet publik.
- Baris resource "aws_route_table_association" menghubungkan subnet yang dibuat dengan tabel routing yang berisi Internet Gateway.
- Dengan ini, instance yang berada di subnet tersebut bisa terhubung ke internet.

**Nat.tf**

```
resource "aws_eip" "latihan-elastic-ip-poco"{
    domain = "vpc"
    depends_on = [ aws_internet_gateway.latihan-igw-poco ]
    tags = {
      Name = "NAT Gateway EIP"
    }
}
```

- Elastic IP ini akan digunakan oleh NAT Gateway.
- Dihubungkan ke Internet Gateway (`depends_on`) agar bisa di-attach setelah IGW ada.

```
resource "aws_nat_gateway" "latihan-nat-gateway-poco" {
  allocation_id = aws_eip.latihan-elastic-ip-poco.id
  subnet_id = aws_subnet.latihan-public-subnet-poco.id

  tags = {
    Name = "latihan nat gateway"
  }
}
```

- NAT Gateway digunakan agar **instance di subnet privat bisa mengakses internet** (misalnya update OS, apt-get, curl API, dll).
- Harus ditaruh di **subnet publik** karena NAT Gateway butuh akses ke internet lewat IGW.

```
resource "aws_route_table" "latihan-private-rt-poco" {
  vpc_id = aws_vpc.latihan-vpc-poco.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_nat_gateway.latihan-nat-gateway-poco.id
  }
  tags = {
    Name = "latihan private rt"
  }
}

resource "aws_route_table_association" "latihan-private_rta-poco" {
  subnet_id = aws_subnet.latihan-private-subnet-poco.id
  route_table_id = aws_route_table.latihan-private-rt-poco.id
}
```

- ini adalah route table khusus untuk subnet privat.
- Route 0.0.0.0/0 mengarah ke NAT Gateway → semua trafik keluar dari subnet privat akan lewat NAT.

**Ec2.tf**

```
resource "aws_security_group" "latihan-security-group-poco2" {
  name = "latihan_sg2"
  vpc_id = aws_vpc.latihan-vpc-poco.id

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

  ingress {
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
    from_port   = 8080
    to_port     = 8080
  }

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

  tags = {
    Name = "Latihan-sg"
  }
}
```

- Security Group adalah firewall yang mengatur lalu lintas masuk dan keluar instance EC2.
- vpc_id id berdasarkan security group yang di buat sebelumnya.
- ingress: o Port 22 untuk SSH agar bisa remote ke EC2. o Port 8080 untuk akses aplikasi berbasis web (misalnya Node.js, Java, dst).
- egress izinkan semua trafik keluar (-1 artinya semua protokol).

```
resource "tls_private_key" "rsa" {
  algorithm = "RSA"
  rsa_bits  = 4096
}

resource "local_file" "LatihanPrivateKeyPairPoco" {
  filename = "latihanKeyPairPoco.pem"
  content  = tls_private_key.rsa.private_key_pem
}

resource "aws_key_pair" "latihanKeyPairPoco2" {
  key_name   = "latihanKeyPairPoco"
  public_key = tls_private_key.rsa.public_key_openssh
}
```

- tls_private_key membuat private key secara lokal (format PEM).
- local_file menyimpan file private key ke file latihanKeyPair di direktori lokal.
- aws_key_pair upload public key-nya ke AWS agar EC2 bisa dikoneksikan via SSH.

```
resource "aws_instance" "latihan-ec2-poco" {
  ami                         = "ami-0815b3f6e070496d4"
  instance_type               = "t2.micro"
  key_name                    = aws_key_pair.latihanKeyPairPoco2.key_name
  vpc_security_group_ids      = [aws_security_group.latihan-security-group-poco2.id]
  subnet_id                   = aws_subnet.latihan-public-subnet-poco.id
  associate_public_ip_address = true

  tags = {
    Name = "Latihan-ec2"
  }
}

resource "aws_instance" "latihan-private-ec2-poco" {
  ami                         = "ami-0815b3f6e070496d4"
  instance_type               = "t2.micro"
  key_name                    = aws_key_pair.latihanKeyPairPoco2.key_name
  vpc_security_group_ids      = [aws_security_group.latihan-security-group-poco2.id]
  subnet_id                   = aws_subnet.latihan-private-subnet-poco.id
  associate_public_ip_address = false

  tags = {
    Name = "Latihan-private-ec2"
  }
}
```

ami ini adalah ID dari Amazon Machine Image (AMI), yaitu sistem operasi yang akan digunakan. "ami-0815b3f6e070496d4" adalah Debian 12 (di Singapura).

- instance_type "t2.micro" jenis instance kecil, masuk ke dalam Free Tier.
- key_name key yang digunakan untuk login via SSH.
- vpc_security_group_ids instance dilindungi oleh security group yang dibuat sebelumnya.
- subnet_id instance ditempatkan di subnet publik yang bisa akses internet.
- user_data file("scriptku.sh") saat EC2 booting pertama kali, akan menjalankan isi file shell script scriptku.sh untuk instalasi otomatis.

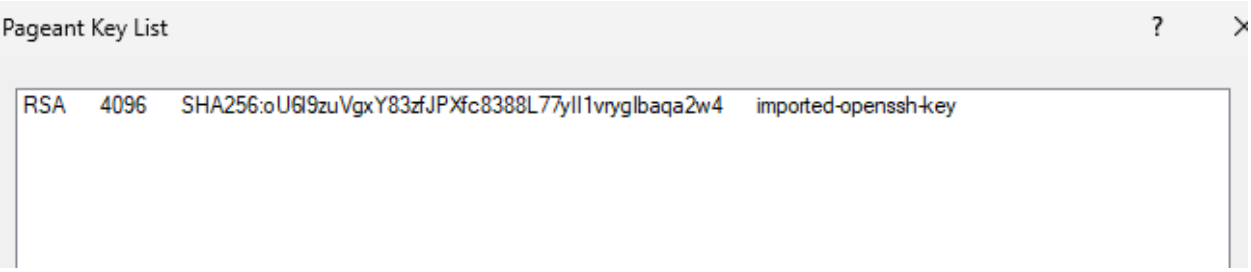
## Running

```
- Reusing previous version of hashicorp/local from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.95.0
- Using previously-installed hashicorp/tls v4.0.6
- Using previously-installed hashicorp/local v2.5.2

erraform has been successfully initialized!

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

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

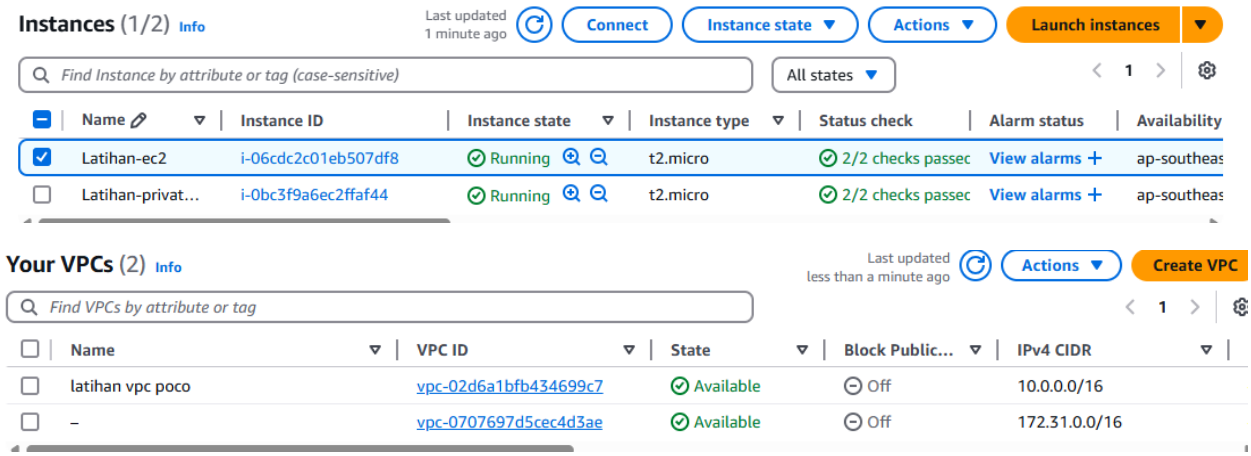
```
PS D:\terraform_project\test\project 8> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_eip.latihan-elastic-ip-poco will be created
  + resource "aws_eip" "latihan-elastic-ip-poco" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = "vpc"
      + id                   = (known after apply)
      + instance             = (known after apply)
      + ipam_pool_id         = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + ptr_record           = (known after apply)
      + public_dns           = (known after apply)
      + public_ip            = (known after apply)
      + public_ipv4_pool     = (known after apply)
      + tags                 = {
```

```
Plan: 3 to add, 1 to change, 1 to destroy.

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_security_group.latihan-security-group-poco2: Destroying... [id=sg-0dbea6706c8ff53b4]
aws_route_table.latihan-private-rt-poco: Modifying... [id=rtb-02529aeb0eaa7b73f]
aws_route_table.latihan-private-rt-poco: Modifications complete after 0s [id=rtb-02529aeb0eaa7b73f]
aws_security_group.latihan-security-group-poco2: Destruction complete after 1s
aws_security_group.latihan-security-group-poco2: Creating...
aws_security_group.latihan-security-group-poco2: Creation complete after 3s [id=sg-07280738ac3951fc3]
aws_instance.latihan-private-ec2-poco: Creating...
aws_instance.latihan-ec2-poco: Creating...
aws_instance.latihan-private-ec2-poco: Still creating... [10s elapsed]
aws_instance.latihan-ec2-poco: Still creating... [10s elapsed]
aws_instance.latihan-ec2-poco: Creation complete after 13s [id=i-0098a0560e5d758a5]
aws_instance.latihan-private-ec2-poco: Creation complete after 13s [id=i-0021c75671295f428]

Apply complete! Resources: 3 added, 1 changed, 1 destroyed.
```

## Peagent




## Hasil



## Subnets (5) Info

Last updated less than a minute ago | Actions | Create subnet

Find subnets by attribute or tag

| Name | Subnet ID | State | VPC | Block Public |
|---|---|---|---|---|
| – | subnet-0dc1da1d5773edb39 | Available | vpc-0707697d5cec4d3ae | Off |
| latihan private subnet | subnet-0cba3bdb3d38c5192 | Available | vpc-02d6a1bfb434699c7 \| latih... | Off |
| – | subnet-0f13d16a29145edbe | Available | vpc-0707697d5cec4d3ae | Off |
| latihan public subnet | subnet-05c64779b994ece8f | Available | vpc-02d6a1bfb434699c7 \| latih... | Off |
| – | subnet-05fe798c82955ca45 | Available | vpc-0707697d5cec4d3ae | Off |

## Internet gateways (2) Info

Actions | Create internet gateway

Find internet gateways by attribute or tag

| Name | Internet gateway ID | State | VPC ID |
|---|---|---|---|
| – | igw-06414a7d6c9ef1a45 | Attached | vpc-0707697d5cec4d3ae |
| latihan igw | igw-0c841595dbec58d8b | Attached | vpc-02d6a1bfb434699c7 \| latihan vpc ... |

## Route tables (4) Info

Last updated 1 minute ago | Actions | Create route table

Find route tables by attribute or tag

| Name | Route table ID | Explicit subnet associ... | Edge associations | Main | VPC |
|---|---|---|---|---|---|
| – | rtb-063b266801aac1680 | – | – | Yes | vpc- |
| latihan private rt | rtb-0113a36b92d49c944 | subnet-0cba3bdb3d38c5... | – | No | vpc- |
| – | rtb-0f62c7a3278dcba83 | – | – | Yes | vpc- |
| latihan public rt | rtb-05f01ee88bf410236 | subnet-05c64779b994ec... | – | No | vpc- |

## Security Groups (1/6) Info

Actions | Export security groups to CSV | Create security group

Find security groups by attribute or tag

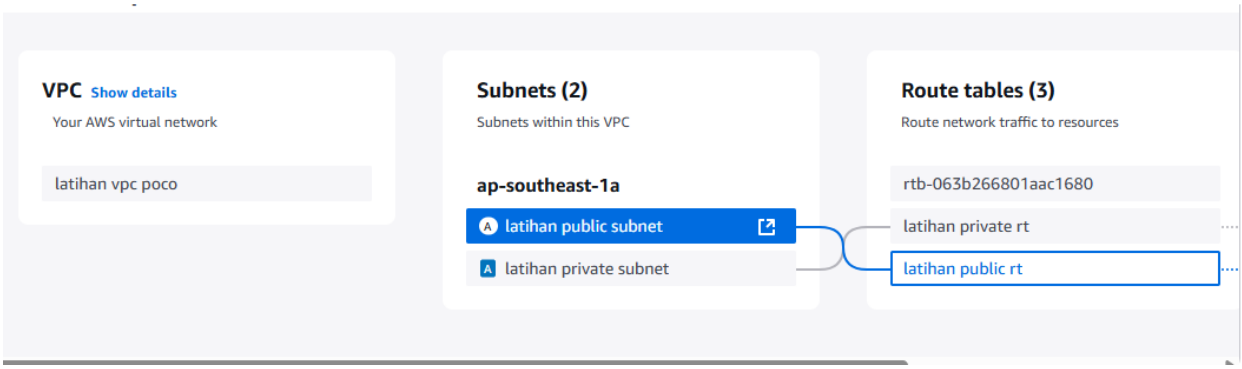
| Name | Security group ID | Security group name | VPC ID |
|---|---|---|---|
| – | sg-0dcfda3b203c65eb3 | launch-wizard-2 | vpc-0707697d5cec4d3ae |
| – | sg-0263f5f32731ba86e | default | vpc-02d6a1bfb434699c7 |
| Latihan-sg | sg-00e385eed3cb5424a | latihan_sg2 | vpc-02d6a1bfb434699c7 |
| latihan-sg | sg-0d1cce6f259e55aaf | latihan_sg | vpc-0707697d5cec4d3ae |

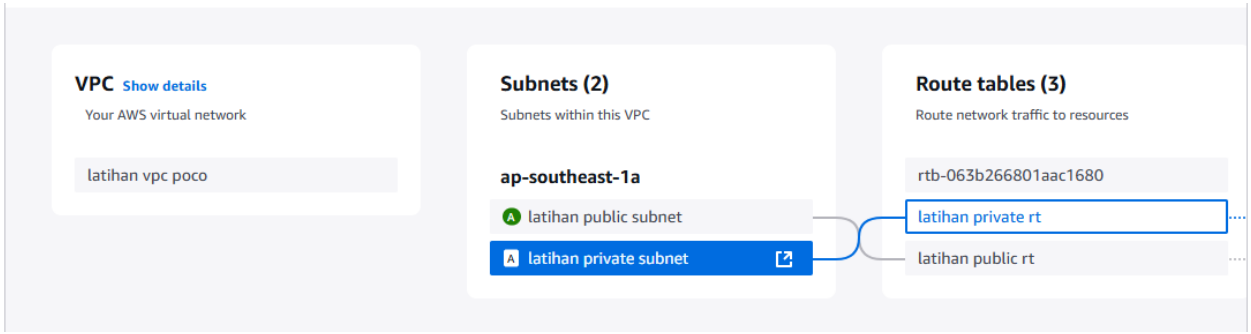
## NAT gateways (1) Info

Actions | Create NAT gateway

Find NAT gateways by attribute or tag

| Name | NAT gateway ID | Connectivity... | State | State message | Primary public I... |
|---|---|---|---|---|---|
| latihan nat gateway | nat-0f1cb0ae11d14b4ea | Public | Available | – | 13.213.120.184 |




```
login as: admin
Authenticating with public key "imported-openssh-key" from agent
Linux ip-10-0-1-57 6.1.0-32-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-
(2025-03-06) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
admin@ip-10-0-1-57:~$ ssh -A admin@10.0.1.57
The authenticity of host '10.0.1.57 (10.0.1.57)' can't be established.
ED25519 key fingerprint is SHA256:bjn3k8P3Vws4COSgAbZUMmgizkRr/PBd26MfF2IcQYc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.57' (ED25519) to the list of known hosts.
Linux ip-10-0-1-57 6.1.0-32-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-
(2025-03-06) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Apr 24 15:28:02 2025 from 114.5.104.77
admin@ip-10-0-1-57:~$
```