LAPORAN PRAKTIKUM AWS

VPC

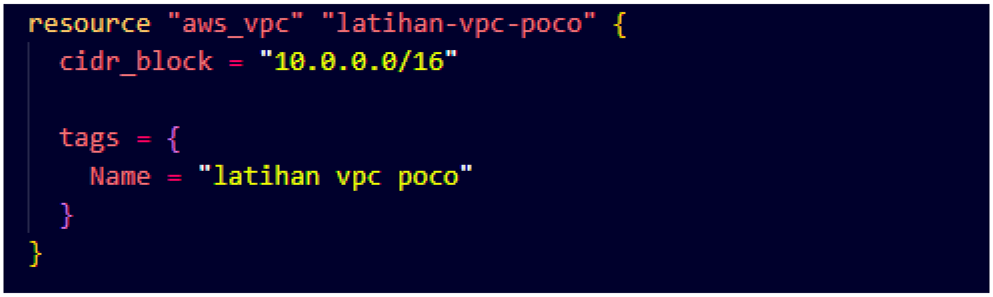
Rahadyan Danang Susetyo Pranawa

Aqil Yoga Pramon

Ahmad Haidar Hafiz

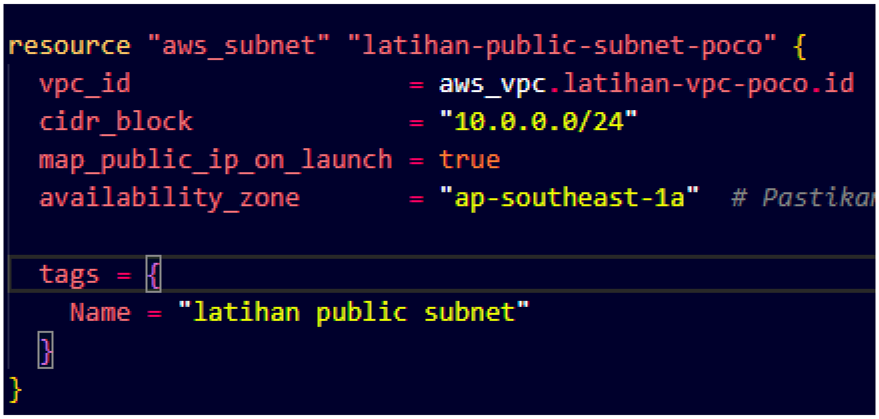
RPA Lexy Mangku S

VPC



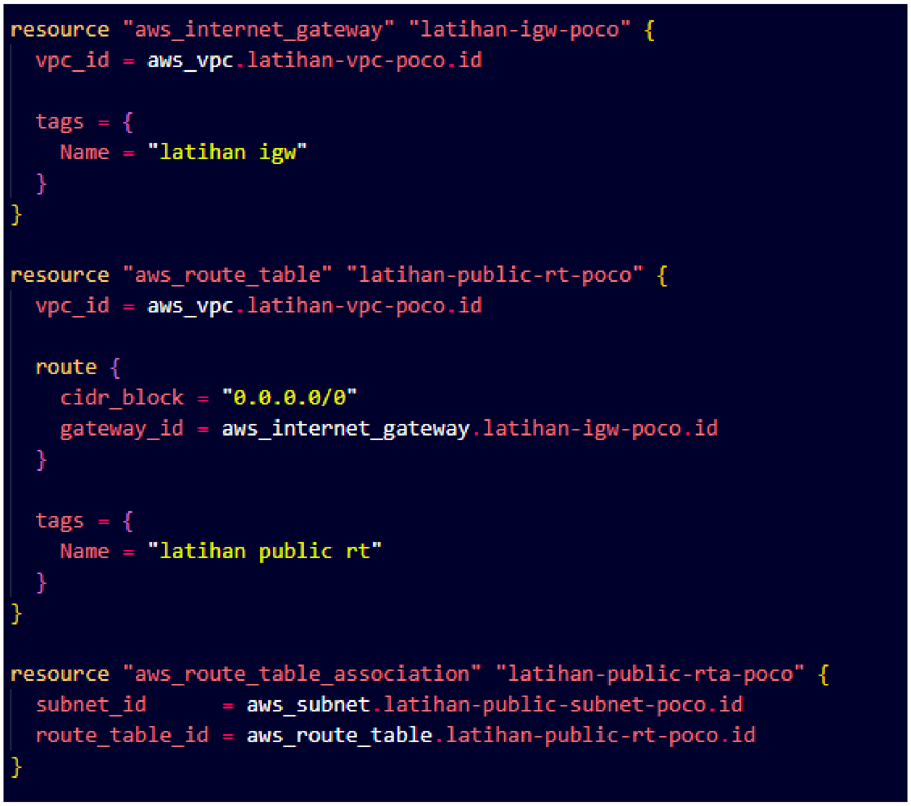
```
resource "aws_vpc" "latihan-vpc-poco" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "latihan vpc poco"
  }
}
```

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.



```
resource "aws_subnet" "latihan-public-subnet-poco" {
  vpc_id                  = aws_vpc.latihan-vpc-poco.id
  cidr_block              = "10.0.0.0/24"
  map_public_ip_on_launch = true
  availability_zone       = "ap-southeast-1a"   # Pastika

  tags = {
    Name = "latihan public subnet"
  }
}
```

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.

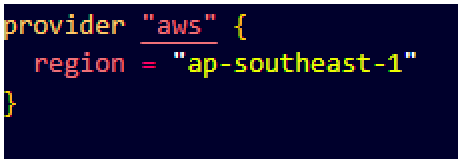
- **`cidr_block = "0.0.0.0/0"`** artinya semua trafik keluar diarahkan ke Internet via Internet Gateway (`gateway_id`).

- Hal ini membuat subnet menjadi **subnet publik**.

- Baris resource "aws_route_table_association" menghubungkan subnet yang dibuat dengan tabel routing yang berisi Internet Gateway.

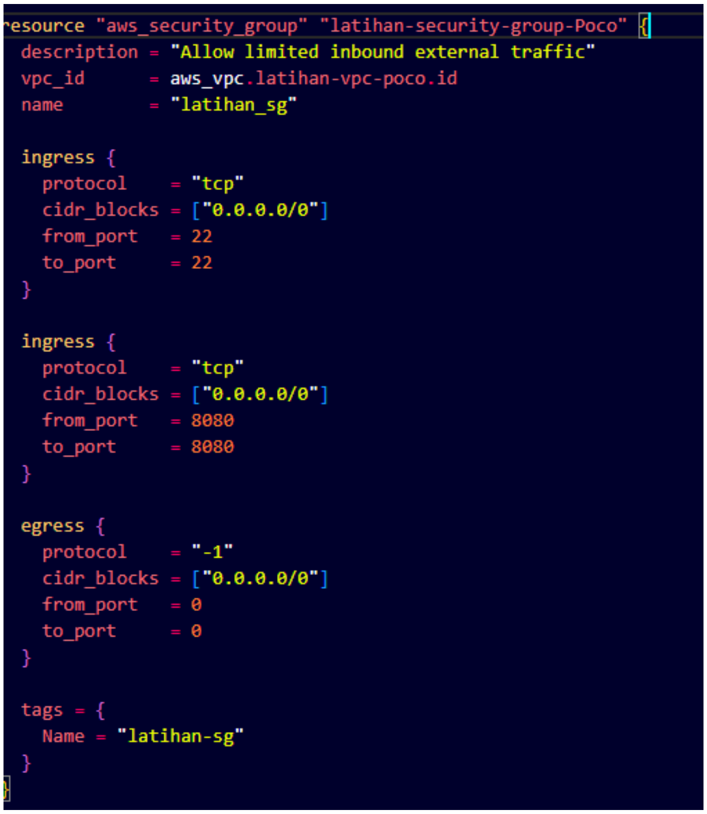
- Dengan ini, instance yang berada di subnet tersebut bisa terhubung ke internet.

MAIN



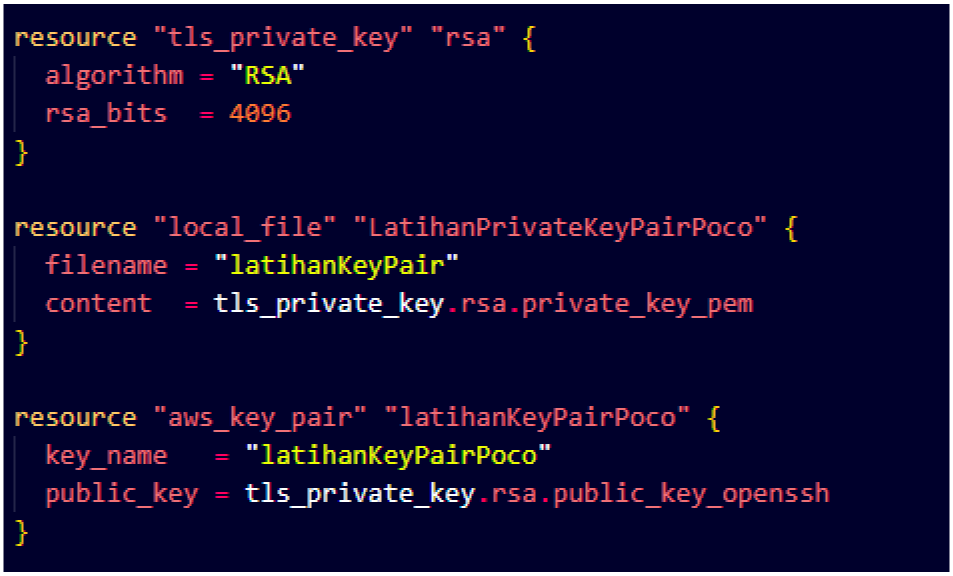
```
provider "aws" {
  region = "ap-southeast-1"
}
```

Memilih region di singapura



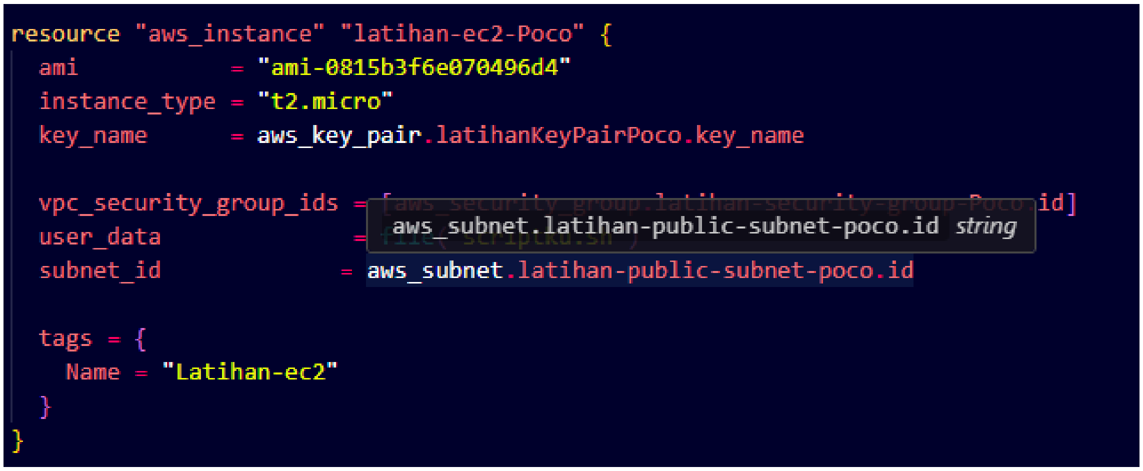
```
resource "aws_security_group" "latihan-security-group-Poco" {
  description = "Allow limited inbound external traffic"
  vpc_id      = aws_vpc.latihan-vpc-poco.id
  name        = "latihan_sg"

  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:
  - Port 22 untuk SSH agar bisa remote ke EC2.
  - 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 = "latihanKeyPair"
  content  = tls_private_key.rsa.private_key_pem
}

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

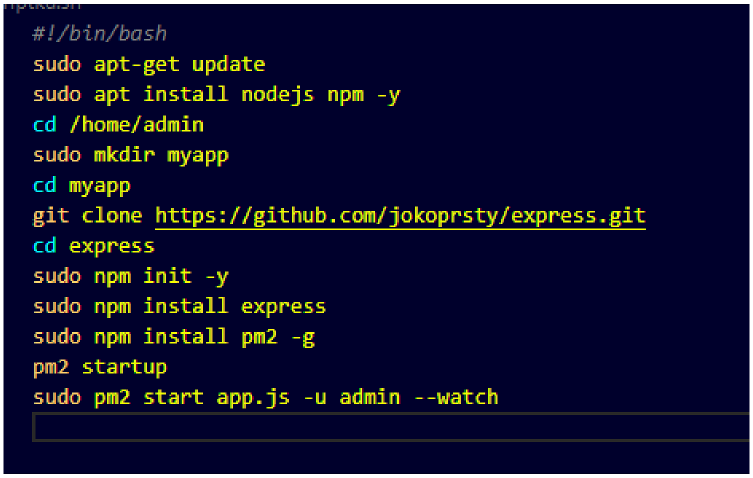
- tls_private_key membuat private key secara lokal (format PEM).
- local_file menyimpan file private key ke file latihanKeyPair di direktori lokal.
- aws_key_pair upload public key-nya ke AWS agar EC2 bisa dikoneksikan via SSH.



```
resource "aws_instance" "latihan-ec2-Poco" {
  ami           = "ami-0815b3f6e070496d4"
  instance_type = "t2.micro"
  key_name      = aws_key_pair.latihanKeyPairPoco.key_name

  vpc_security_group_ids = [aws_security_group.latihan-security-group-Poco.id]
  user_data              = file("scriptku.sh")
  subnet_id              = aws_subnet.latihan-public-subnet-poco.id

  tags = {
    Name = "Latihan-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.

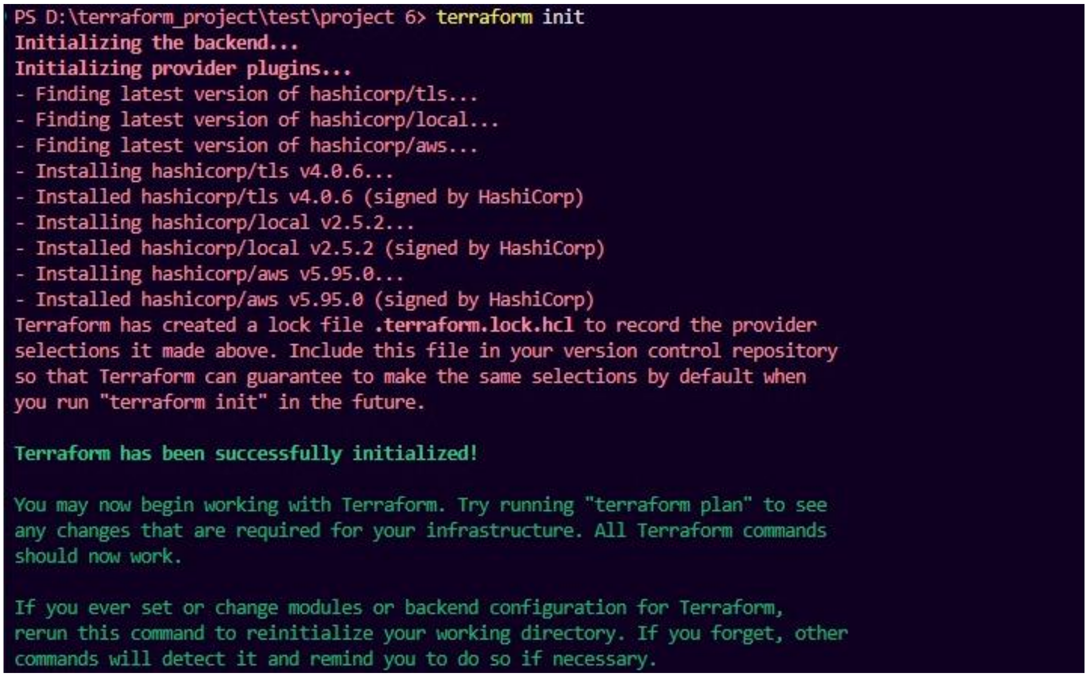
SCRIPTKU



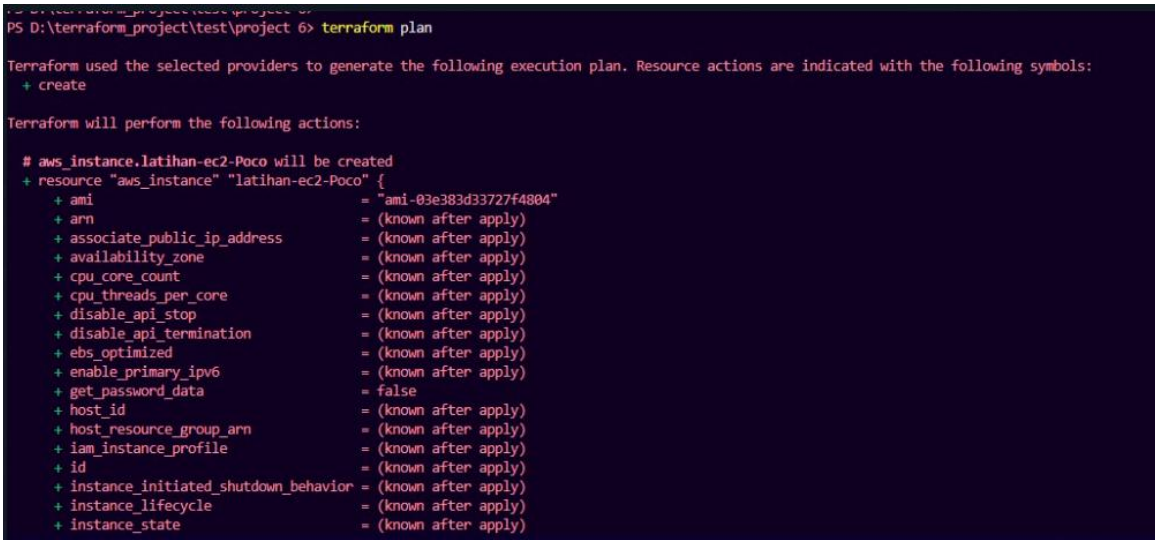
```
#!/bin/bash
sudo apt-get update
sudo apt install nodejs npm -y
cd /home/admin
sudo mkdir myapp
cd myapp
git clone https://github.com/jokoprsty/express.git
cd express
sudo npm init -y
sudo npm install express
sudo npm install pm2 -g
pm2 startup
sudo pm2 start app.js -u admin --watch
```

Hasil

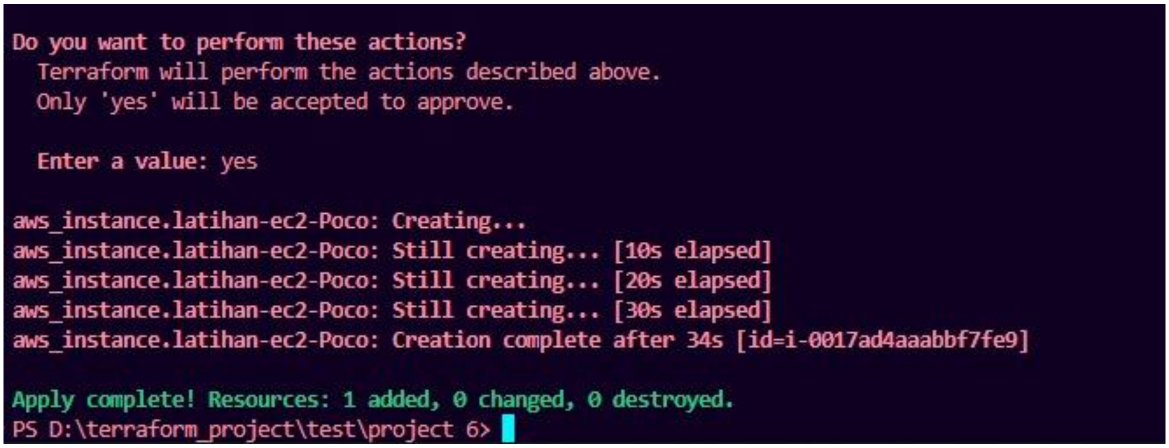
Jalankan terraform



```
PS D:\terraform_project\test\project 6> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/tls...
- Finding latest version of hashicorp/local...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/tls v4.0.6...
- Installed hashicorp/tls v4.0.6 (signed by HashiCorp)
- Installing hashicorp/local v2.5.2...
- Installed hashicorp/local v2.5.2 (signed by HashiCorp)
- Installing hashicorp/aws v5.95.0...
- Installed hashicorp/aws v5.95.0 (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.
```

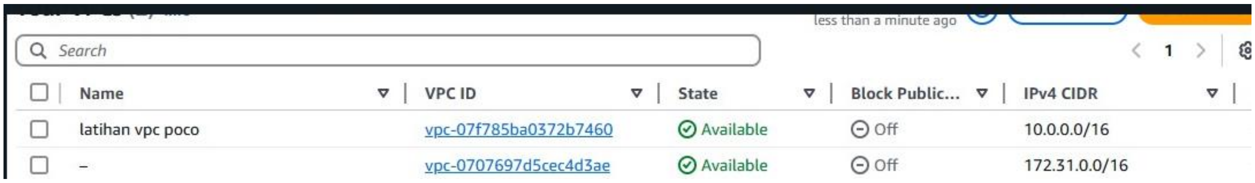


```
PS D:\terraform_project\test\project 6> terraform plan

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_instance.latihan-ec2-Poco will be created
  + resource "aws_instance" "latihan-ec2-Poco" {
      + ami                                  = "ami-03e383d33727f4804"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
```

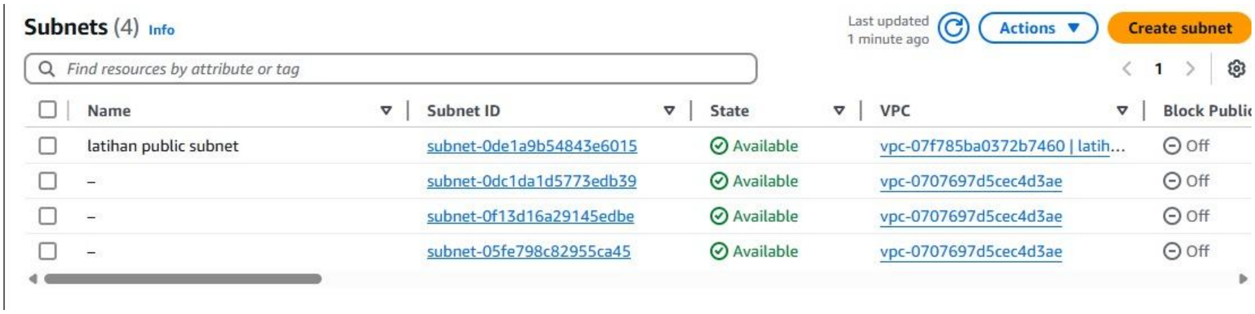


```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.latihan-ec2-Poco: Creating...
aws_instance.latihan-ec2-Poco: Still creating... [10s elapsed]
aws_instance.latihan-ec2-Poco: Still creating... [20s elapsed]
aws_instance.latihan-ec2-Poco: Still creating... [30s elapsed]
aws_instance.latihan-ec2-Poco: Creation complete after 34s [id=i-0017ad4aaabbf7fe9]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS D:\terraform_project\test\project 6>
```

Cek bagian vpc

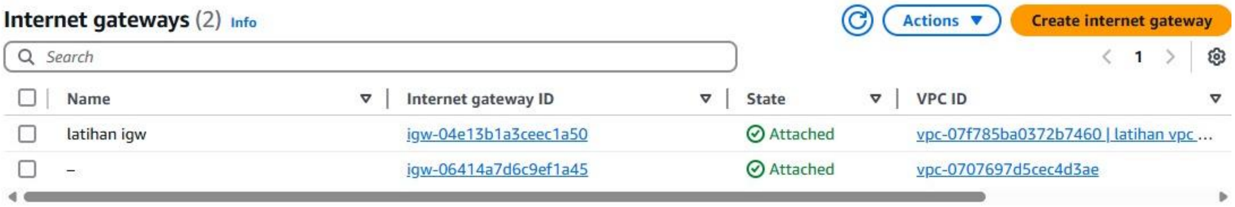


□	Name	VPC ID	State	Block Public...	IPv4 CIDR
□	latihan vpc poco	vpc-07f785ba0372b7460	✓ Available	⊖ Off	10.0.0.0/16
□	–	vpc-0707697d5cec4d3ae	✓ Available	⊖ Off	172.31.0.0/16

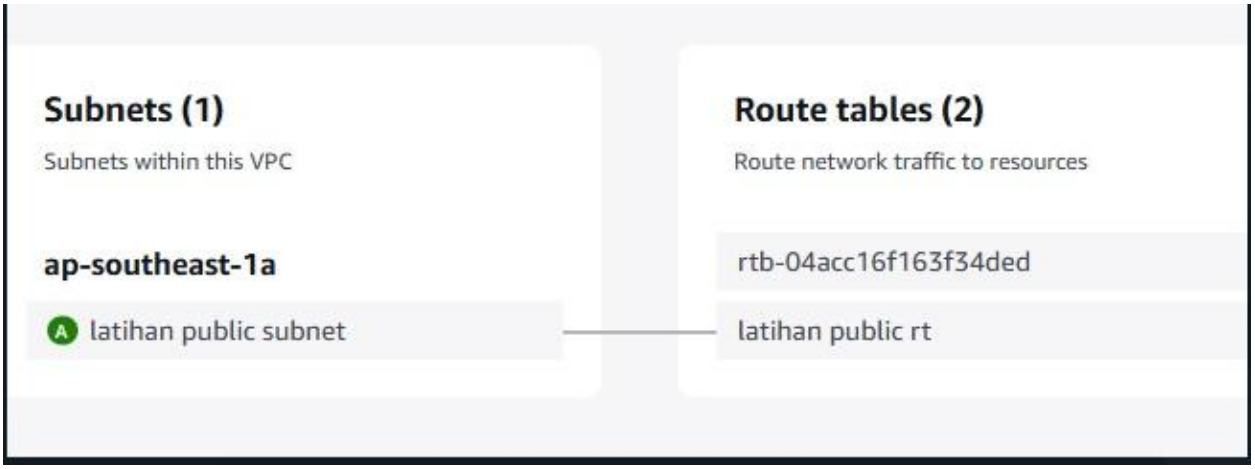


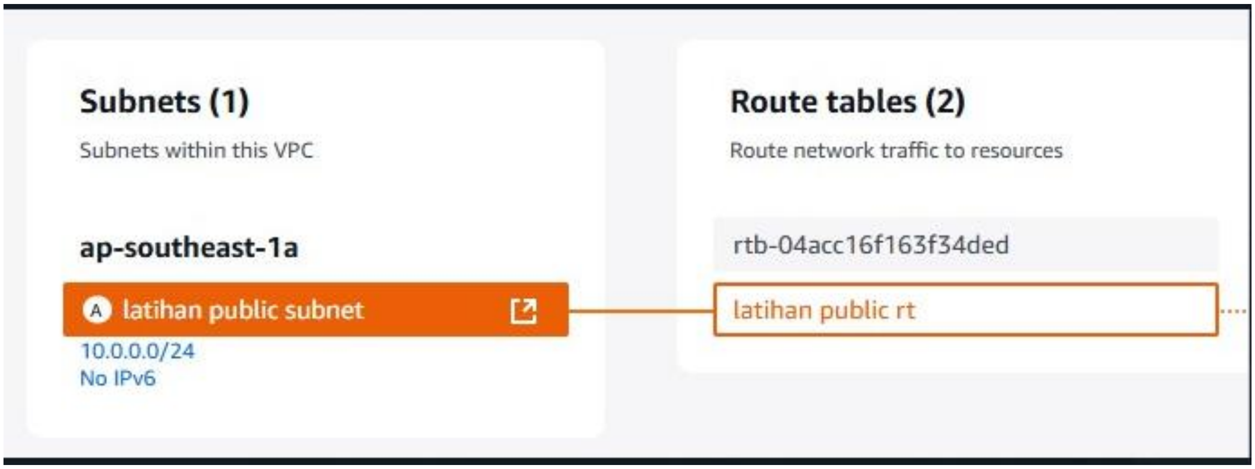
Subnets (4) Info

□	Name	Subnet ID	State	VPC	Block Public
□	latihan public subnet	subnet-0de1a9b54843e6015	✓ Available	vpc-07f785ba0372b7460 | latih...	⊖ Off
□	–	subnet-0dc1da1d5773edb39	✓ Available	vpc-0707697d5cec4d3ae	⊖ Off
□	–	subnet-0f13d16a29145edbe	✓ Available	vpc-0707697d5cec4d3ae	⊖ Off
□	–	subnet-05fe798c82955ca45	✓ Available	vpc-0707697d5cec4d3ae	⊖ Off

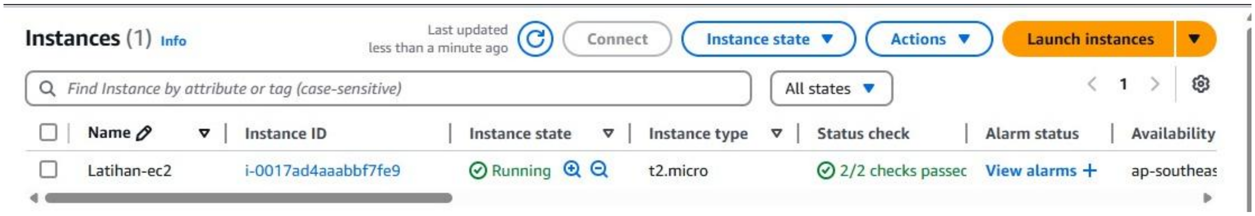


☐	Name	Internet gateway ID	State	VPC ID
☐	latihan igw	igw-04e13b1a3ceec1a50	✓ Attached	vpc-07f785ba0372b7460 | latihan vpc ...
☐	–	igw-06414a7d6c9ef1a45	✓ Attached	vpc-0707697d5cec4d3ae



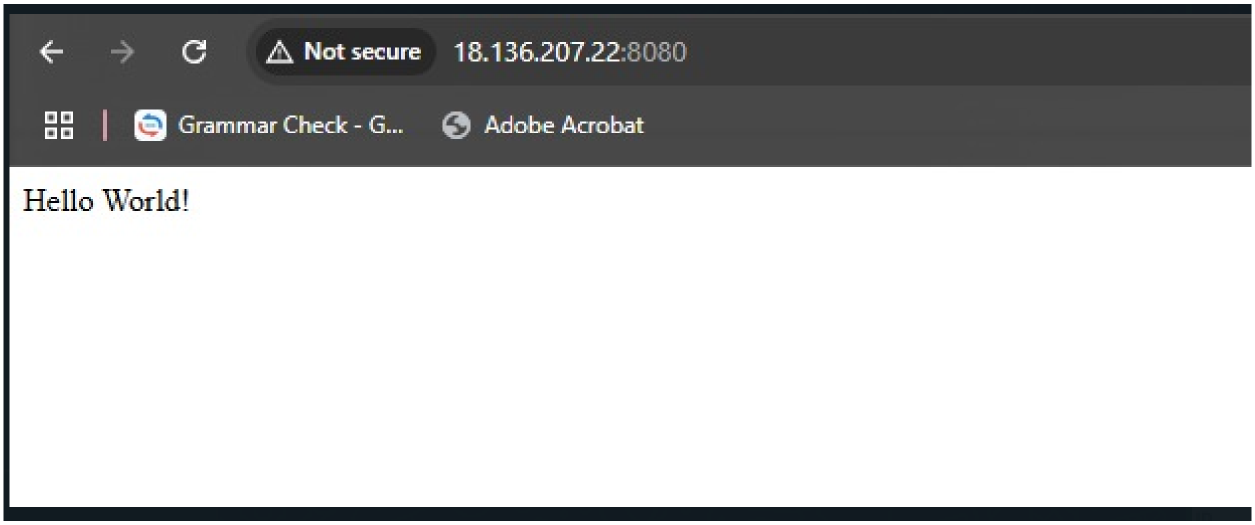


Cek Instances



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
☐	Latihan-ec2	i-0017ad4aaabbf7fe9	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-southeas

Jalankan pada web



Hello World!