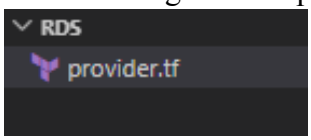
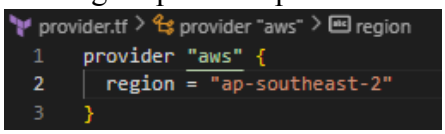
Laporan Pertemuan  11 Workshop Aplikasi dan Komputasi Awan
A HAIDAR HAFIZ
R.P.A. LEXY MANGKU SAPUTRA
RAHADYAN DANANG SUSETYO PRAMONO
AQIL YOGA PRAMONO

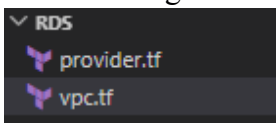
Buat file dengan nama provider.tf



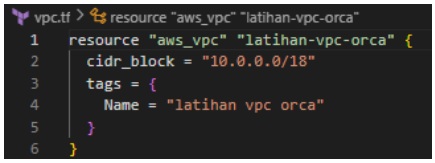

Atur region pada file provider.tf




Buat file dengan nama vpc.tf




Buat vpc dengan nama yang diinginkan




Buat access control list untuk mengatur lalu lintas subnet

vpc.tf > resource "aws_network_acl" "latihan-acl-orca"

```
resource "aws_network_acl" "latihan-acl-orca" {
  vpc_id = aws_vpc.latihan-vpc-orca.id
  ingress {
    protocol = "tcp"
    rule_no = 100
    action = "allow"
    cidr_block = "0.0.0.0/0"
    from_port = 3306
    to_port = 3306
  }
  ingress {
    protocol = "tcp"
    rule_no = 101
    action = "allow"
    cidr_block = "0.0.0.0/0"
    from_port = 22
    to_port = 22
  }

  egress {
    protocol = -1
    rule_no = 100
    action = "allow"
    cidr_block = "0.0.0.0/0"
    from_port = 0
    to_port = 0
  }
  tags = {
    Name = "latihan acl orca"
  }
}
```

Membuat zona A subnet

vpc.tf > resource "aws_subnet" "latihan-public-subnet-orca"

```
40  resource "aws_subnet" "latihan-public-subnet-orca" {
41    vpc_id = aws_vpc.latihan-vpc-orca.id
42    cidr_block = "10.0.0.0/24"
43    map_public_ip_on_launch = "true"
44    availability_zone = "ap-southeast-2a"
45    tags = {
46      Name = "latihan public subnet orca"
47    }
48  }
49
50  resource "aws_subnet" "latihan-private-subnet-orca" {
51    vpc_id = aws_vpc.latihan-vpc-orca.id
52    cidr_block = "10.0.1.0/24"
53    map_public_ip_on_launch = "false"
54    availability_zone = "ap-southeast-2a"
55    tags = {
56      Name = "latihan private subnet orca"
57    }
58  }
```

Membuat zona B subnet

vpc.tf > resource "aws_subnet" "latihan-private-subnet-orca2"

```
60  resource "aws_subnet" "latihan-private-subnet-orca2" {
61    vpc_id = aws_vpc.latihan-vpc-orca.id
62    cidr_block = "10.0.2.0/24"
63    map_public_ip_on_launch = "false"
64    availability_zone = "ap-southeast-2b"
65    tags = {
66      Name = "latihan private subnet orca2"
67    }
68  }
```

Membuat internet gateway

vpc.tf > resource "aws_internet_gateway" "latihan-igw-orca"

```
70  resource "aws_internet_gateway" "latihan-igw-orca" {
71    vpc_id = aws_vpc.latihan-vpc-orca.id
72    tags = {
73      Name = "latihan igw orca"
74    }
75  }
```

Membuat association access control list

vpc.tf > resource "aws_network_acl_association" "latihan-acl-assoc-orca"

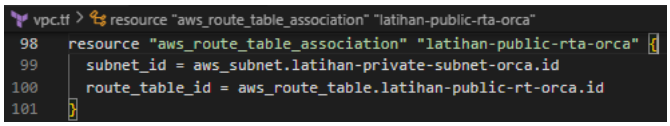
```
77  resource "aws_network_acl_association" "latihan-acl-assoc-orca" {
78    network_acl_id = aws_network_acl.latihan-acl-orca.id
79    subnet_id = aws_subnet.latihan-private-subnet-orca.id
80  }
81
82  resource "aws_network_acl_association" "latihan-acl-assoc-orca2" {
83    network_acl_id = aws_network_acl.latihan-acl-orca.id
84    subnet_id = aws_subnet.latihan-private-subnet-orca2.id
85  }
```

Membuat route table public

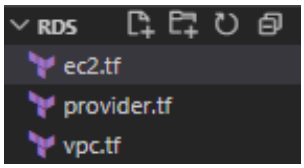
vpc.tf > resource "aws_route_table" "latihan-public-rt-orca"

```
87  resource "aws_route_table" "latihan-public-rt-orca" {
88    vpc_id = aws_vpc.latihan-vpc-orca.id
89    route {
90      cidr_block = "0.0.0.0/0"
91      gateway_id = aws_internet_gateway.latihan-igw-orca.id
92    }
93    tags = {
94      Name = "latihan publice rt orca"
95    }
96  }
```

Membuat association public dan router public




Buat file dengan nama ec2.tf




Buat security group

ec2.tf > resource "aws_security_group" "latihan-security-group-orca"

```
resource "aws_security_group" "latihan-security-group-orca" {
  description = "Allow limited inbound external traffic"
  vpc_id = "${aws_vpc.latihan-vpc-orca.id}"
  name = "latihan-sg-orca"
  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 = 3000
    to_port = 3000
  }
  ingress {
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
    from_port = 3006
    to_port = 3006
  }
  ingress {
    protocol = "icmp"
    cidr_blocks = ["0.0.0.0/0"]
    from_port = -1
    to_port = -1
  }
  egress {
    protocol = -1
    cidr_blocks = ["0.0.0.0/0"]
    from_port = 0
    to_port = 0
  }
  tags = {
    Name = "latihan-sg-orca"
  }
}
```

Buat key pa

ec2.tf > resource "tls_private_key" "rsa-orca"

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

resource "local_file" "LatihanPrivateKeyPairOrca" {
  filename = "LatihanPrivateKeyPair"
  content = tls_private_key.rsa-orca.private_key_pem
}

resource "aws_key_pair" "LatihanPublicKeyPair" {
  key_name = "LatihanPublicKeyPair"
  public_key = tls_private_key.rsa-orca.public_key_openssh
}
```

Buat data untuk digunakan memasukkan variable ke file scriptku.sh

ec2.tf > data "template_file" "user_data_orca"

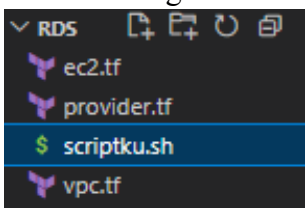
```
data "template_file" "user_data_orca" {
  template = "${file("scriptku.sh")}"
  vars = {
    rds_address = "${aws_db_instance.latihan_db_rds.address}"
    rds_username = "${aws_db_instance.latihan_db_rds.username}"
    rds_password = "${aws_db_instance.latihan_db_rds.password}"
    rds_db_name = "${aws_db_instance.latihan_db_rds.db_name}"
  }
}
```

Buat instance

ec2.tf > resource "aws_instance" "latihan-ec2"

```
resource "aws_instance" "latihan-ec2" {
  depends_on = [ aws_db_instance.latihan_db_rds ]
  ami = "ami-08ea97f528e500b62"
  instance_type = "t2.micro"
  key_name = aws_key_pair.LatihanPublicKeyPair.key_name
  vpc_security_group_ids = ["${aws_security_group.latihan-security-group-orca.id}"]
  subnet_id = "${aws_subnet.latihan-public-subnet-orca.id}"
  #user_data = "${file("scriptku.sh")}"
  user_data = data.template_file.user_data_orca.rendered
  tags = {
    Name = "Latihan-ec2-orca"
  }
}
```

Scriptku
Buat file dengan nama scriptku.sh

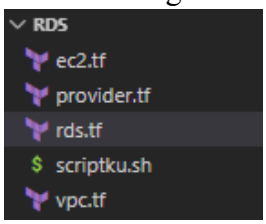



Isi dari file scriptku.sh

scriptku.sh

```
#!/bin/bash
sudo apt-get update
sudo apt install default-mysql-client-y
#sudo mysql -u "${rds_username}" --password="${rds_password}" --host "${rds_address}" -e "CREATE DATABASE ${rds_db_name} /*\!40100 DEFAULT CHARACTER SET utf8 /*;"
sudo apt install nodejs npm -y
cd /home/admin
sudo mkdir myapp
cd myapp
git clone https://github.com/jokoprsty/latihan_rds.git
cd latihan_rds
echo "DB_USER=${rds_username}" >> .env
echo "DB_PASS=${rds_password}" >> .env
echo "DB_NAME=${rds_db_name}" >> .env
echo "DB_HOST=${rds_address}" >> .env
sudo npm install
sudo npm install pm2 -g
sudo pm2 start /home/admin/myapp/latihan_rds/app.js -u admin --watch
sudo pm2 save
pm2 startup
```

RDS
Buat file dengan nama rds.tf




Buat subnet group rds

rds.tf > resource "aws_db_subnet_group" "latihan_subnet_db"

```
resource "aws_db_subnet_group" "latihan_subnet_db" {
  name = "latihan_subnet_db"
  subnet_ids = [aws_subnet.latihan-private-subnet-orca,aws_subnet.latihan-private-subnet-orca2.id]
  tags = {
    Name = "latihan_subnet_db"
  }
}
```

Buat security group untuk rds

rds.tf > resource "aws_security_group" "rds"

```
resource "aws_security_group" "rds" {
  name = "latihan_rds_sg-orca"
  vpc_id = aws_vpc.latihan-vpc-orca.id
  ingress {
    from_port = 3306
    to_port = 3306
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port = 3306
    to_port = 3306
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  tags = {
    Name = "latihan_rds_sg"
  }
}
```

Buat instance rds

rds.tf > resource "aws_db_instance" "latihan_db_rds"

```
resource "aws_db_instance" "latihan_db_rds" {
  identifier = "latihan-db-rds"
  instance_class = "db.t3.micro"
  allocated_storage = 20
  engine = "mariadb"
  engine_version = "10.6.14"
  username = "latihan"
  password = 12345678
  db_name = "my_project"
  db_subnet_group_name = aws_db_subnet_group.latihan_subnet_db.name
  vpc_security_group_ids = [aws_security_group.rds.id]
  publicly_accessible = false
  skip_final_snapshot = true
}
```

Menjalankan Terraform
Jalankan “terraform init” di command prompt direktori project

```
C:\Users\ibamq\OneDrive\Documents\Kuliah\Semester 4\Workshop Aplikasi Dan Komputasi Awan\Pertemuan ke-11\rds>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/template...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/local...
- Finding latest version of hashicorp/tls...
- Installing hashicorp/template v2.2.0...
- Installed hashicorp/template v2.2.0 (signed by HashiCorp)
- Installing hashicorp/aws v5.97.0...
- Installed hashicorp/aws v5.97.0 (signed by HashiCorp)
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
- Installing hashicorp/tls v4.1.0...
- Installed hashicorp/tls v4.1.0 (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.
```

Jika “terraform init” berhasil dijalankan, selanjutnya jalankan “terraform plan”

```
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

  # data.template_file.user_data_orca will be read during apply
  # (config refers to values not yet known)
 <= data "template_file" "user_data_orca" {
      + id       = (known after apply)
      + rendered = (known after apply)
      + template = <<-EOT
            #!/bin/bash
            sudo apt-get update
            sudo apt install default-mysql-client-y
            #sudo mysql -u "${rds_username}" --password="${rds_password}" --host "${rds_address}" -e "CREATE DATABASE ${rds_db_na
me} /*\!40100 DEFAULT CHARACTER SET utf8 /*;"
            sudo apt install nodejs npm -y
            cd /home/admin
            sudo mkdir myapp
            cd myapp
            git clone https://github.com/jokoprsty/latihan_rds.git
            cd latihan_rds
            echo "DB_USER=${rds_username}" >> .env
            echo "DB_PASS=${rds_password}" >> .env
            echo "DB_NAME=${rds_db_name}" >> .env
            echo "DB_HOST=${rds_address}" >> .env
            sudo npm install
            sudo npm install pm2 -g
            sudo pm2 start /home/admin/myapp/latihan_rds/app.js -u admin --watch
            sudo pm2 save
```

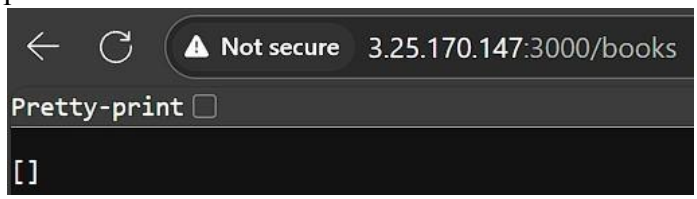
```
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "LatihanPrivateKeyPair"
      + id                   = (known after apply)
    }

  # tls_private_key.rsa-orca will be created
  + resource "tls_private_key" "rsa-orca" {
      + algorithm                   = "RSA"
      + ecdsa_curve                 = "P224"
      + id                          = (known after apply)
      + private_key_openssh         = (sensitive value)
      + private_key_pem             = (sensitive value)
      + private_key_pem_pkcs8       = (sensitive value)
      + public_key_fingerprint_md5  = (known after apply)
      + public_key_fingerprint_sha256 = (known after apply)
      + public_key_openssh          = (known after apply)
      + public_key_pem              = (known after apply)
      + rsa_bits                    = 4096
    }

Plan: 18 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run
"terraform apply" now.
```

Ketika “terraform plan” berhasil, maka jalankan “terraform apply”

```
C:\Users\ibamq\OneDrive\Documents\Kuliah\Semester 4\Workshop Aplikasi Dan Komputasi Awan\Pertemuan ke-11\rds>terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

  # data.template_file.user_data_orca will be read during apply
  # (config refers to values not yet known)
 <= data "template_file" "user_data_orca" {
      + id       = (known after apply)
      + rendered = (known after apply)
      + template = <<-EOT
            #!/bin/bash
            sudo apt-get update
            sudo apt install default-mysql-client-y
            #sudo mysql -u "${rds_username}" --password="${rds_password}" --host "${rds_address}" -e "CREATE DATABASE ${rds_db_na
me} /*\!40100 DEFAULT CHARACTER SET utf8 /*;"
            sudo apt install nodejs npm -y
            cd /home/admin
            sudo mkdir myapp
            cd myapp
            git clone https://github.com/jokoprsty/latihan_rds.git
            cd latihan_rds
            echo "DB_USER=${rds_username}" >> .env
            echo "DB_PASS=${rds_password}" >> .env
            echo "DB_NAME=${rds_db_name}" >> .env
            echo "DB_HOST=${rds_address}" >> .env
            sudo npm install
            sudo npm install pm2 -g
```

```
aws_subnet.latihan-public-subnet-orca: Still creating... [10s elapsed]
aws_subnet.latihan-public-subnet-orca: Creation complete after 13s [id=subnet-0c789f735b7c5c39e]
aws_db_instance.latihan_db_rds: Still creating... [10s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [20s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [30s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [40s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [50s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [1m0s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [1m10s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [1m20s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [1m30s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [1m40s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [1m50s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [2m0s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [2m10s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [2m20s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [2m30s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [2m40s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [2m50s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [3m0s elapsed]
aws_db_instance.latihan_db_rds: Still creating... [3m10s elapsed]
aws_db_instance.latihan_db_rds: Creation complete after 3m17s [id=db-YLFXYG73XZDEGEKZOVJCKMXH74]
data.template_file.user_data_orca: Reading...
data.template_file.user_data_orca: Read complete after 0s [id=caa249a624d1365e3b857bb346e3b94e59381a0454445cf4903e8b0290f1b0c5]
aws_instance.latihan-ec2: Creating...
aws_instance.latihan-ec2: Still creating... [10s elapsed]
aws_instance.latihan-ec2: Still creating... [20s elapsed]
aws_instance.latihan-ec2: Creation complete after 26s [id=i-0f6b702bdbb3b5aff]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.
```

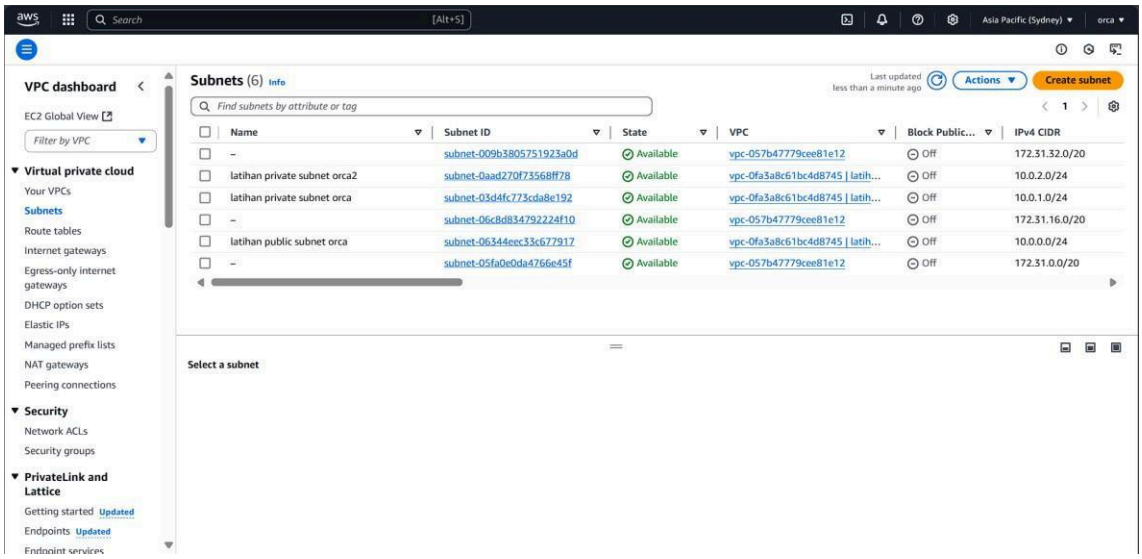
Tugas
Buka browser, kemudian copy alamat ipv4 public ec2 di aws ke browser, buka dengan port 3000



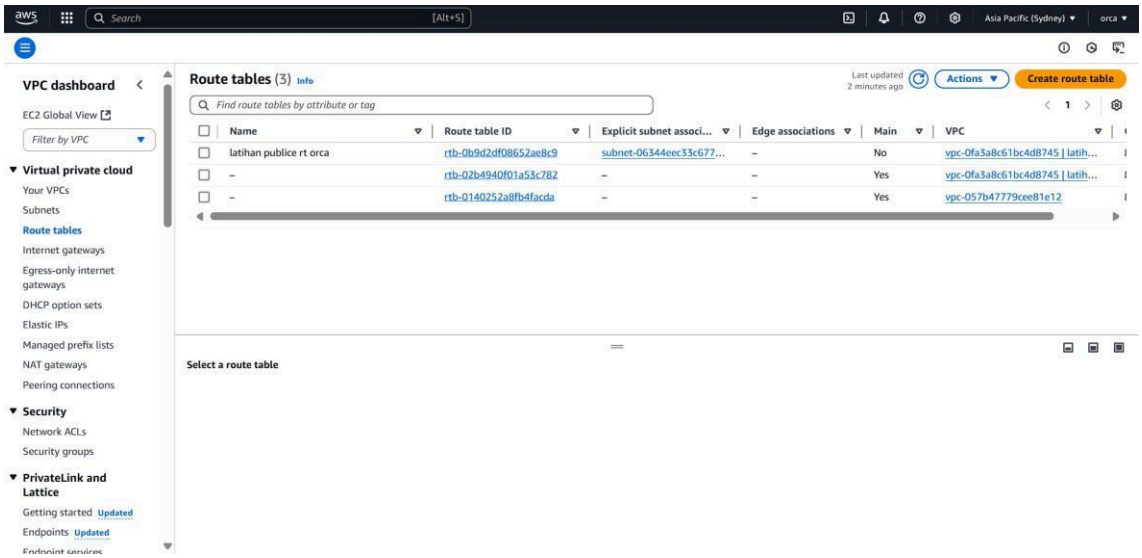

Insert data kedalam mysql melalui ec2(gunakan ssh), insert data dengan menggunakan alamat mysql yang ada di /home/admin/myapp/latihan_rds/.env, kemudian buka di browser

```
MariaDB [my_project]> INSERT INTO Books SET title="Buku 1",author="Author 1",createdAt=NOW(),updatedAt=NOW();
Query OK, 1 row affected (0.002 sec)
```

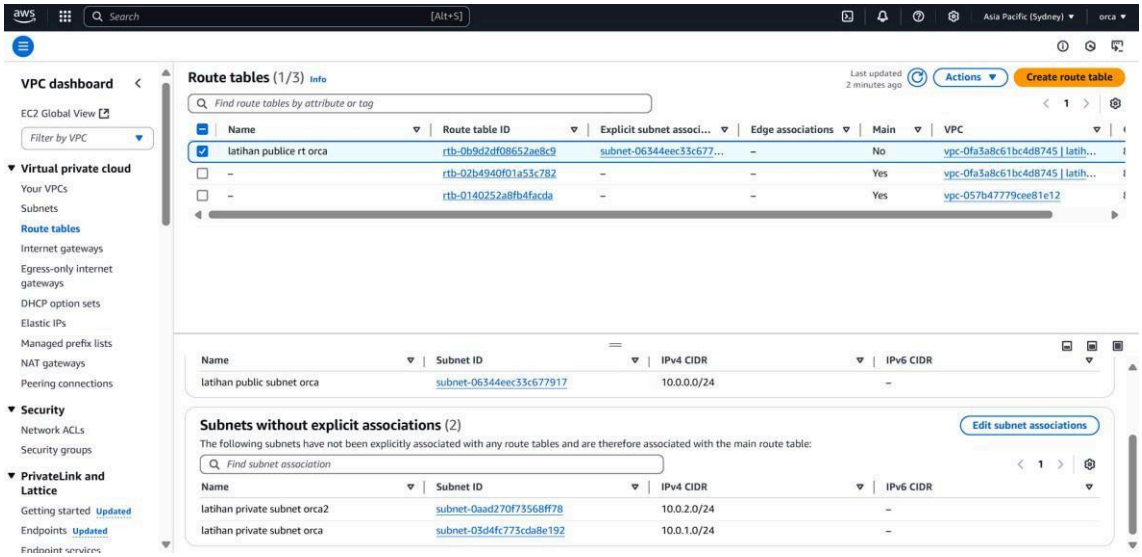
Screenshot VPC
Subnet



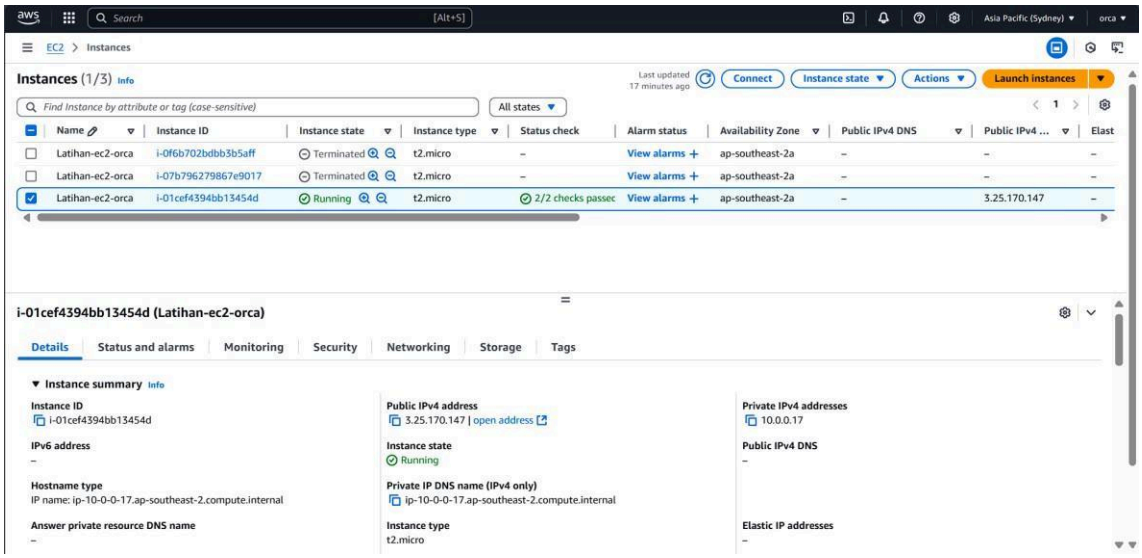

Route



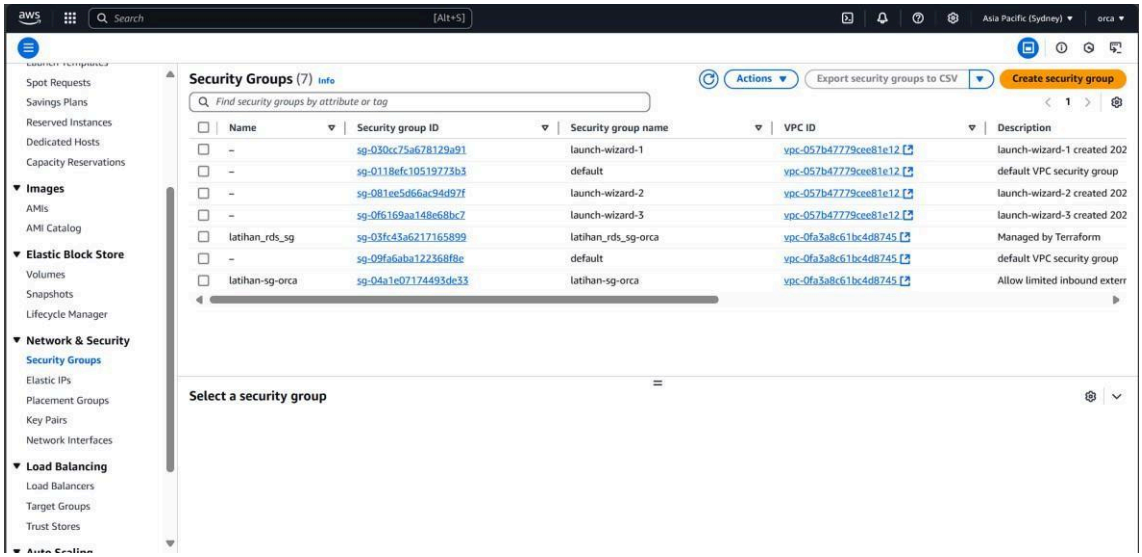

Route association



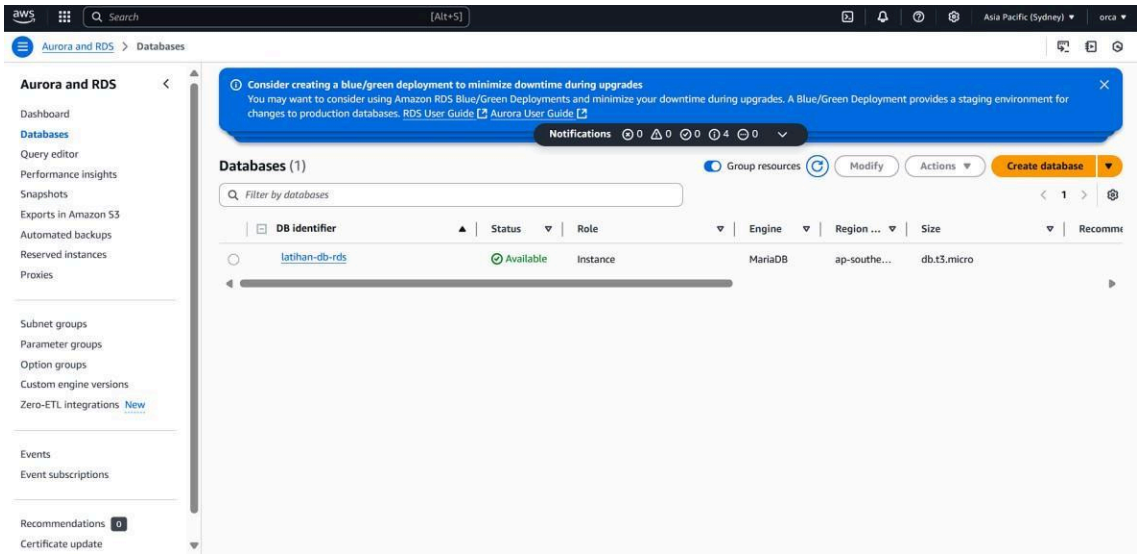
Screenshot EC2
Instance



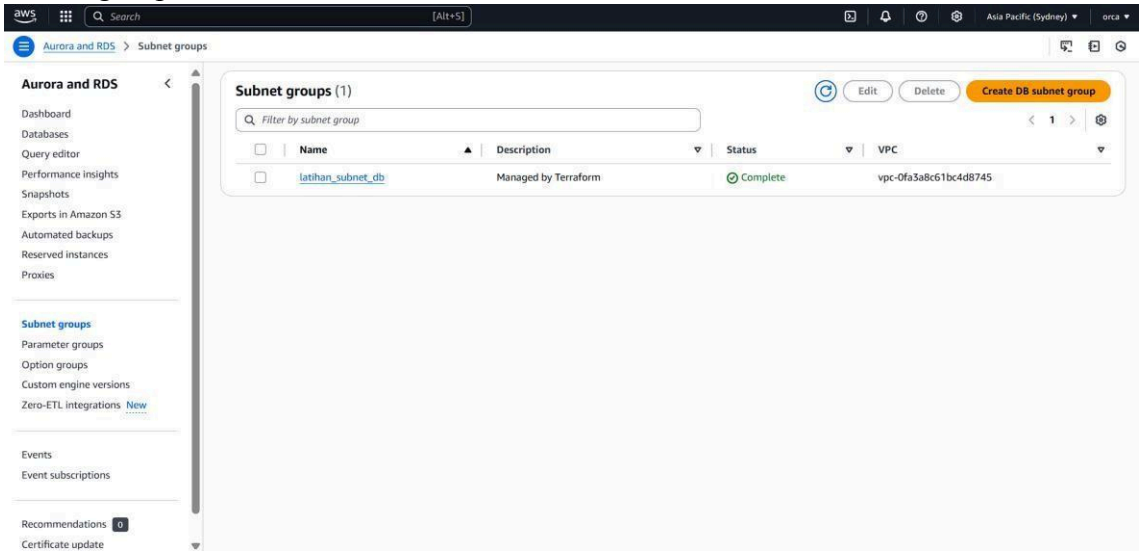

Security group



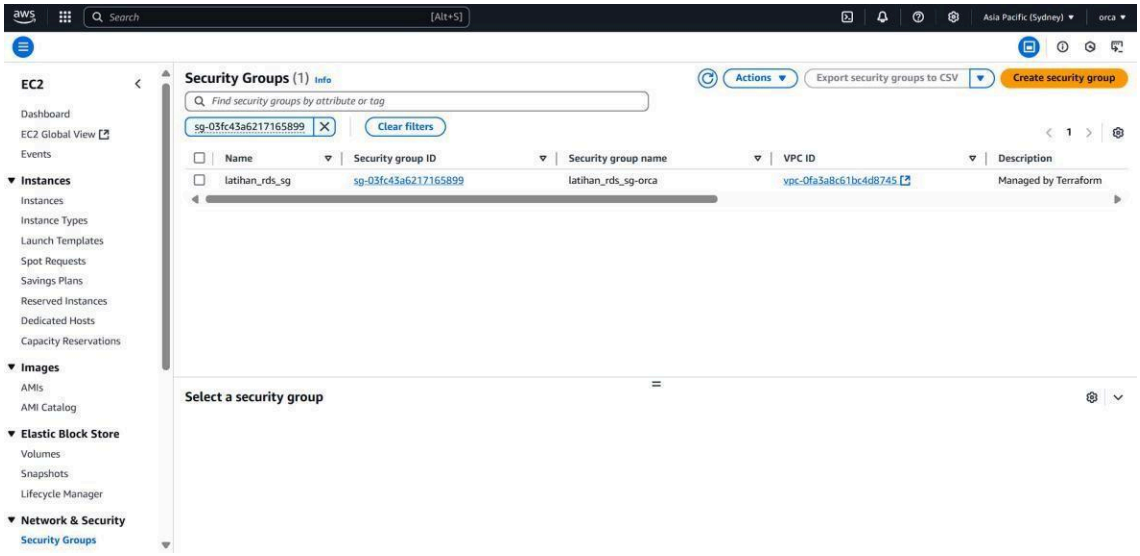

Screenshot RDS
Database


aws
Search
[Alt+S]
Asia Pacific (Sydney)
orca
Aurora and RDS > Databases
Aurora and RDS
Dashboard
Databases
Query editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 0
Certificate update
Consider creating a blue/green deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide
Notifications 0 0 0 4 0
Databases (1)
Group resources
Modify
Actions
Create database
Filter by databases
1
DB identifier
Status
Role
Engine
Region ...
Size
Recomme
latihan-db-rds
Available
Instance
MariaDB
ap-southe...
db.t3.micro

Subnet group




Security group




Terraform destroy

```
C:\Users\ibamq\OneDrive\Documents\Kuliah\Semester 4\Workshop Aplikasi Dan Komputasi Awan\Pertemuan ke-11\rds>terraform destroy
tls_private_key.rsa-orca: Refreshing state... [id=8c8641a1643c501bffb698ae21e84bffbe61cd81]
local_file.LatihanPrivateKeyPairOrca: Refreshing state... [id=b4d59058633f691f8ae0fae8f69386cb4b4add73]
aws_key_pair.LatihanPublicKeyPair: Refreshing state... [id=LatihanKeyPair]
aws_vpc.latihan-vpc-orca: Refreshing state... [id=vpc-0fa3a8c61bc4d8745]
aws_internet_gateway.latihan-igw-orca: Refreshing state... [id=igw-03ecff5036b56e6d7]
aws_subnet.latihan-private-subnet-orca2: Refreshing state... [id=subnet-0aad270f73568ff78]
aws_subnet.latihan-public-subnet-orca: Refreshing state... [id=subnet-06344eec33c677917]
aws_subnet.latihan-private-subnet-orca: Refreshing state... [id=subnet-03d4fc773cda8e192]
aws_security_group.rds: Refreshing state... [id=sg-03fc43a6217165899]
aws_network_acl.latihan-acl-orca: Refreshing state... [id=acl-03c88746723ec0ad1]
aws_security_group.latihan-security-group-orca: Refreshing state... [id=sg-04a1e07174493de33]
aws_db_subnet_group.latihan_subnet_db: Refreshing state... [id=latihan_subnet_db]
aws_route_table.latihan-public-rt-orca: Refreshing state... [id=rtb-0b9d2df08652ae8c9]
aws_network_acl_association.latihan-acl-assoc-orca2: Refreshing state... [id=aclassoc-0fa2ab34593031ba6]
aws_network_acl_association.latihan-acl-assoc-orca: Refreshing state... [id=aclassoc-095c2825b2fc72425]
aws_route_table_association.latihan-public-rta-orca: Refreshing state... [id=rtbassoc-032a0717304ef1f67]
aws_db_instance.latihan_db_rds: Refreshing state... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M]
data.template_file.user_data_orca: Reading...
data.template_file.user_data_orca: Read complete after 0s [id=caa249a624d1365e3b857bb346e3b94e59381a0454445cf4903e8b0290f1b0c5]
aws_instance.latihan-ec2: Refreshing state... [id=i-01cef4394bb13454d]

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

Terraform will perform the following actions:

  # aws_db_instance.latihan_db_rds will be destroyed
  - resource "aws_db_instance" "latihan_db_rds" {
      - address                             = "latihan-db-rds.cxyakw2ycnfi.ap-southeast-2.rds.amazonaws.com" -> null
      - allocated_storage                   = 20 -> null
```

```
aws_security_group.latihan-security-group-orca: Destruction complete after 2s
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 10s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 20s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 30s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 40s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 50s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 1m0s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 1m10s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 1m20s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 1m30s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 1m40s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 1m50s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 2m0s elapsed]
aws_db_instance.latihan_db_rds: Still destroying... [id=db-GIFEWXMYDVN2VEYBDMVKMRLA4M, 2m10s elapsed]
aws_db_instance.latihan_db_rds: Destruction complete after 2m13s
aws_db_subnet_group.latihan_subnet_db: Destroying... [id=latihan_subnet_db]
aws_security_group.rds: Destroying... [id=sg-03fc43a6217165899]
aws_db_subnet_group.latihan_subnet_db: Destruction complete after 1s
aws_subnet.latihan-private-subnet-orca2: Destroying... [id=subnet-0aad270f73568ff78]
aws_subnet.latihan-private-subnet-orca: Destroying... [id=subnet-03d4fc773cda8e192]
aws_security_group.rds: Still destroying... [id=sg-03fc43a6217165899, 10s elapsed]
aws_subnet.latihan-private-subnet-orca2: Still destroying... [id=subnet-0aad270f73568ff78, 10s elapsed]
aws_subnet.latihan-private-subnet-orca: Still destroying... [id=subnet-03d4fc773cda8e192, 10s elapsed]
aws_subnet.latihan-private-subnet-orca: Destruction complete after 12s
aws_subnet.latihan-private-subnet-orca2: Destruction complete after 12s
aws_security_group.rds: Destruction complete after 13s
aws_vpc.latihan-vpc-orca: Destroying... [id=vpc-0fa3a8c61bc4d8745]
aws_vpc.latihan-vpc-orca: Destruction complete after 1s

Destroy complete! Resources: 18 destroyed.
```