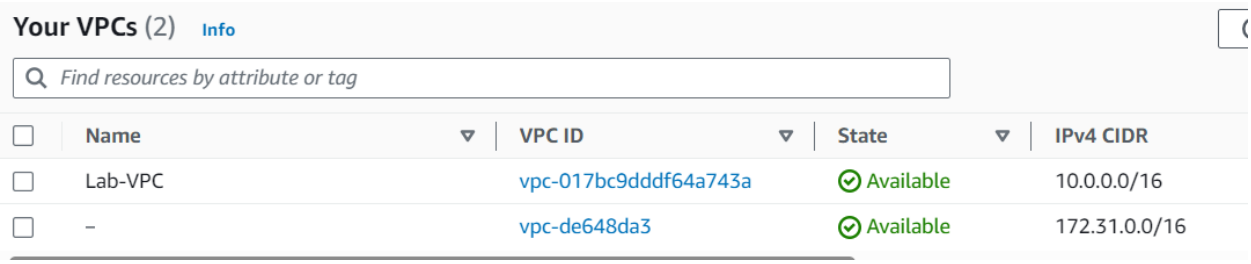
Region = us-east-1

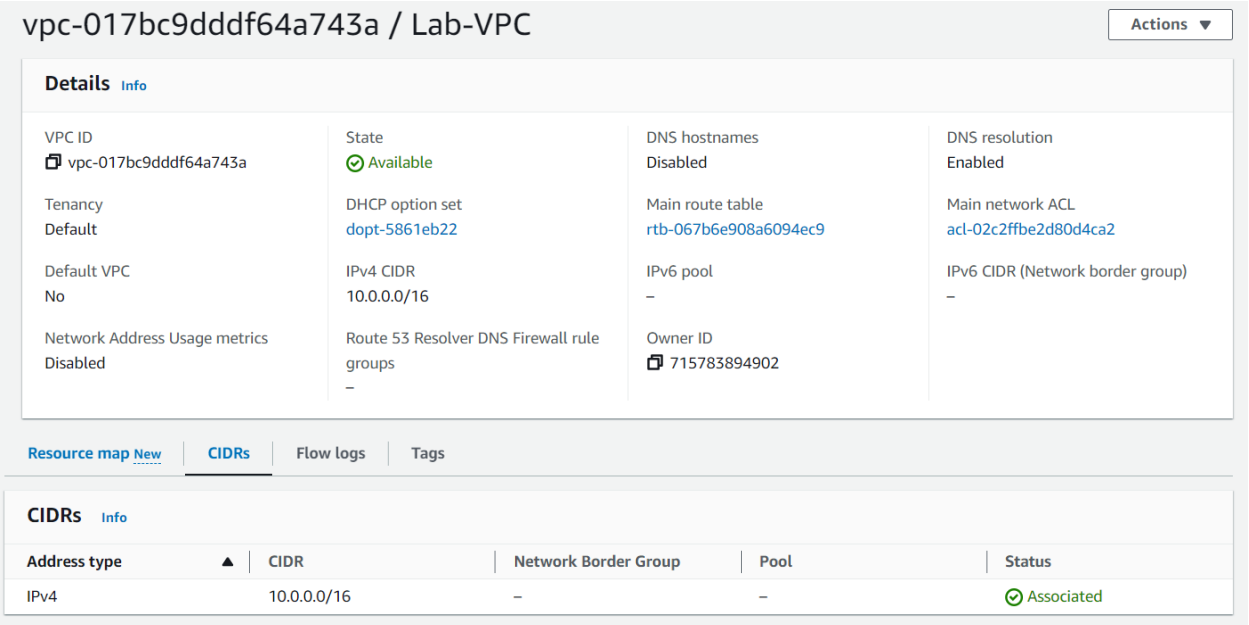
VPC

Code:

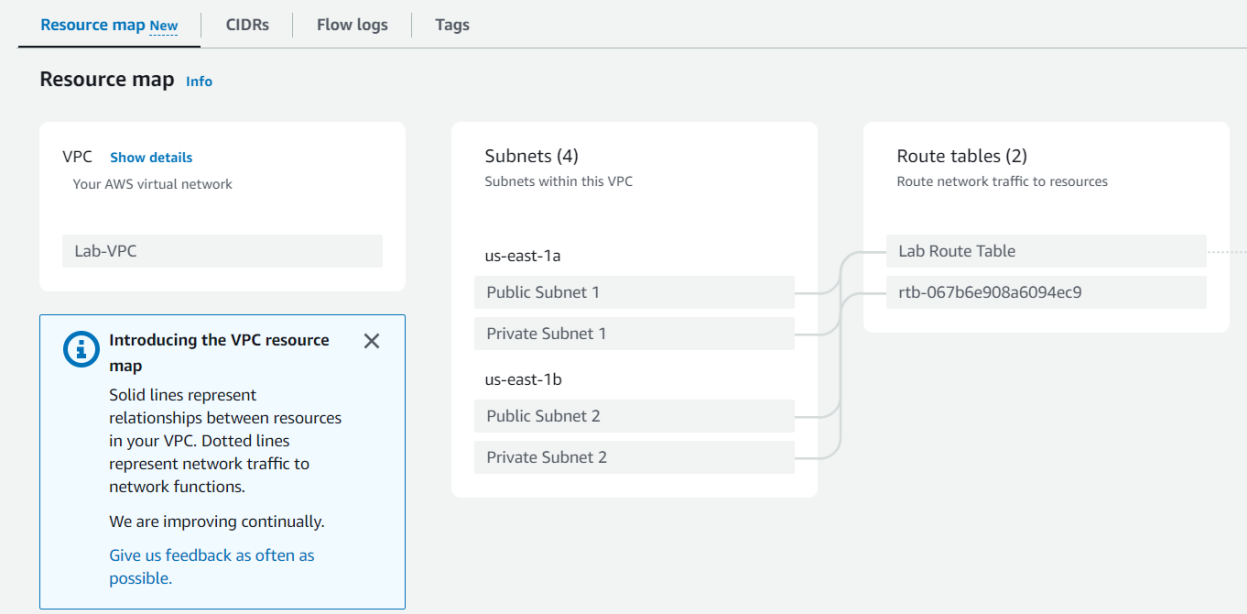
```
resource "aws_vpc" "lab" {
  cidr_block       = "10.0.0.0/16"
  instance_tenancy = "default"

  tags = {
    Name = "Lab-VPC"
  }
}
```

Result:





Resource Map (check next steps for setup of subnets and route tables)



Subnets

Code:

```
resource "aws_subnet" "public_subnet1" {
 vpc_id     = aws_vpc.lab.id
 cidr_block = "10.0.0.0/24"
 availability_zone = "us-east-1a"

 tags = {
  Name = "Public Subnet 1"
 }
}

resource "aws_subnet" "public_subnet2" {
 vpc_id     = aws_vpc.lab.id
 cidr_block = "10.0.1.0/24"
 availability_zone = "us-east-1b"

 tags = {
  Name = "Public Subnet 2"
 }
}
```

Result:

Subnets (10) Info

Actions | Create subnet

Find resources by attribute or tag

1

Name	Subnet ID	State	VPC	IPv4 CIDR
Public Subnet 1	subnet-07e76d2effbdd1048	Available	vpc-017bc9dddf64a743a | Lab-...	10.0.0.0/24
–	subnet-c9f62de8	Available	vpc-de648da3	172.31.80.0/20
Public Subnet 2	subnet-0d2285c7f14149ec8	Available	vpc-017bc9dddf64a743a | Lab-...	10.0.1.0/24
–	subnet-f97c10b4	Available	vpc-de648da3	172.31.16.0/20
–	subnet-6ef16760	Available	vpc-de648da3	172.31.64.0/20
Private Subnet 1	subnet-0890c71dc174daccb	Available	vpc-017bc9dddf64a743a | Lab-...	10.0.2.0/24
–	subnet-86a37ad9	Available	vpc-de648da3	172.31.32.0/20
–	subnet-e634d4d7	Available	vpc-de648da3	172.31.48.0/20
–	subnet-275e8341	Available	vpc-de648da3	172.31.0.0/20
Private Subnet 2	subnet-0d408a138187bf56c	Available	vpc-017bc9dddf64a743a | Lab-...	10.0.3.0/24

Subnets (1/10) Info

Actions | Create subnet

Find resources by attribute or tag

1

Name	Subnet ID	State	VPC	IPv4 CIDR
Public Subnet 1	subnet-07e76d2effbdd1048	Available	vpc-017bc9dddf64a743a | Lab-...	10.0.0.0/24
–	subnet-c9f62de8	Available	vpc-de648da3	172.31.80.0/20

Subnet ID
subnet-07e76d2effbdd1048

Available IPv4 addresses
250

Network border group
us-east-1

Default subnet
No

Customer-owned IPv4 pool
–

IPv6-only
No

DNS64
Disabled

Subnet ARN
arn:aws:ec2:us-east-1:715783894902:subnet/subnet-07e76d2effbdd1048

IPv6 CIDR
–

VPC
vpc-017bc9dddf64a743a | Lab-VPC

Auto-assign public IPv4 address
No

Outpost ID
–

Hostname type
IP name

State
Available

Availability Zone
us-east-1a

Route table
rtb-09f83c48ac045e565 | Lab Route Table

Auto-assign IPv6 address
No

IPv4 CIDR reservations
–

Resource name DNS A record
Disabled

IPv4 CIDR
10.0.0.0/24

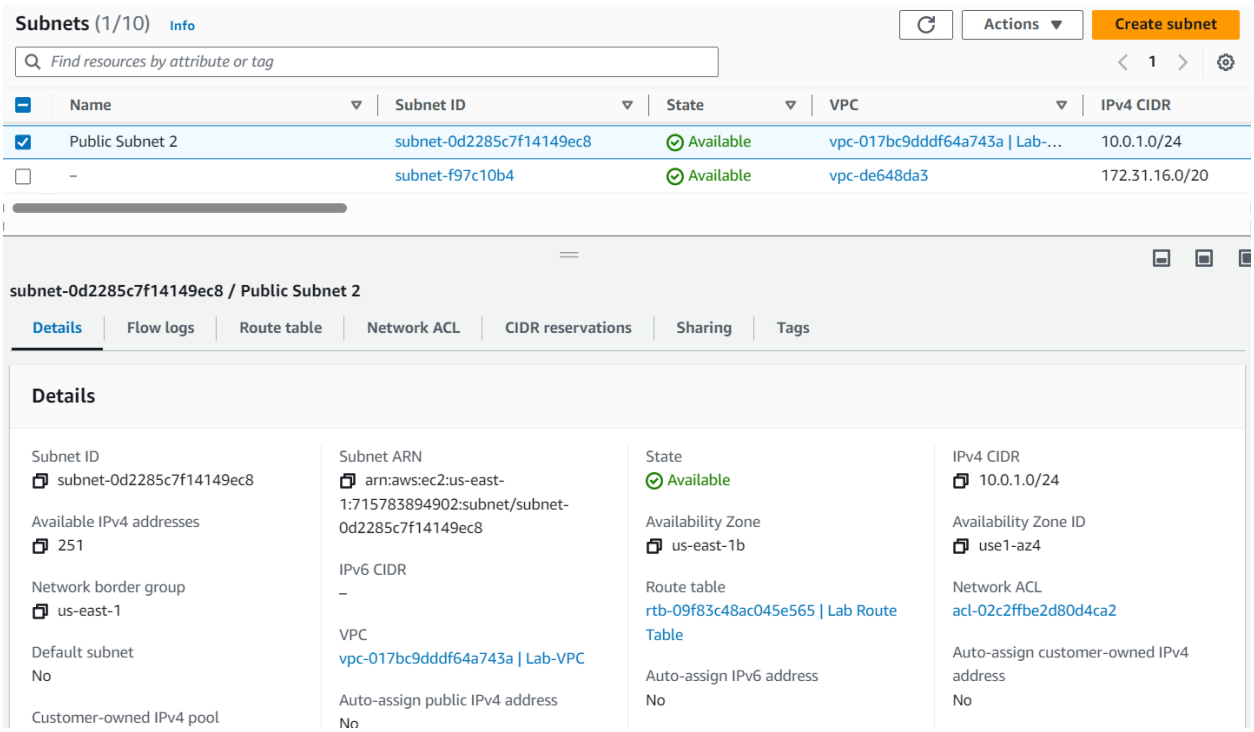
Availability Zone ID
use1-az2

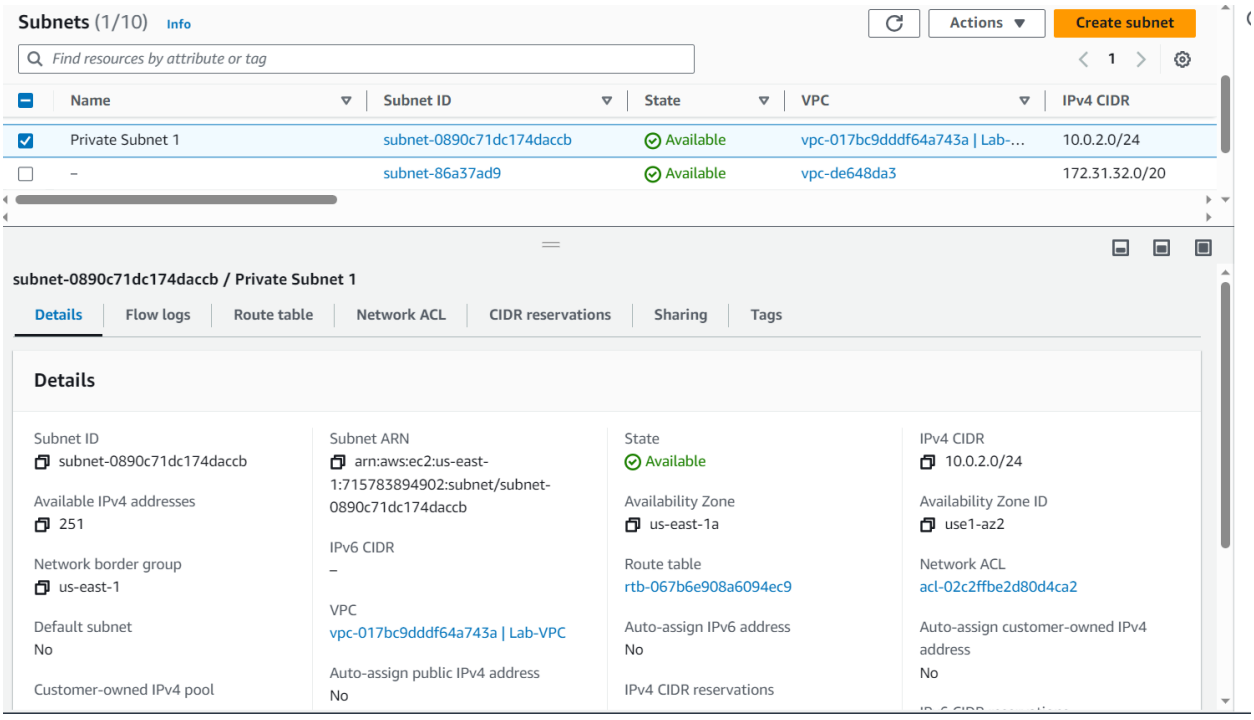
Network ACL
acl-02c2ffbe2d80d4ca2

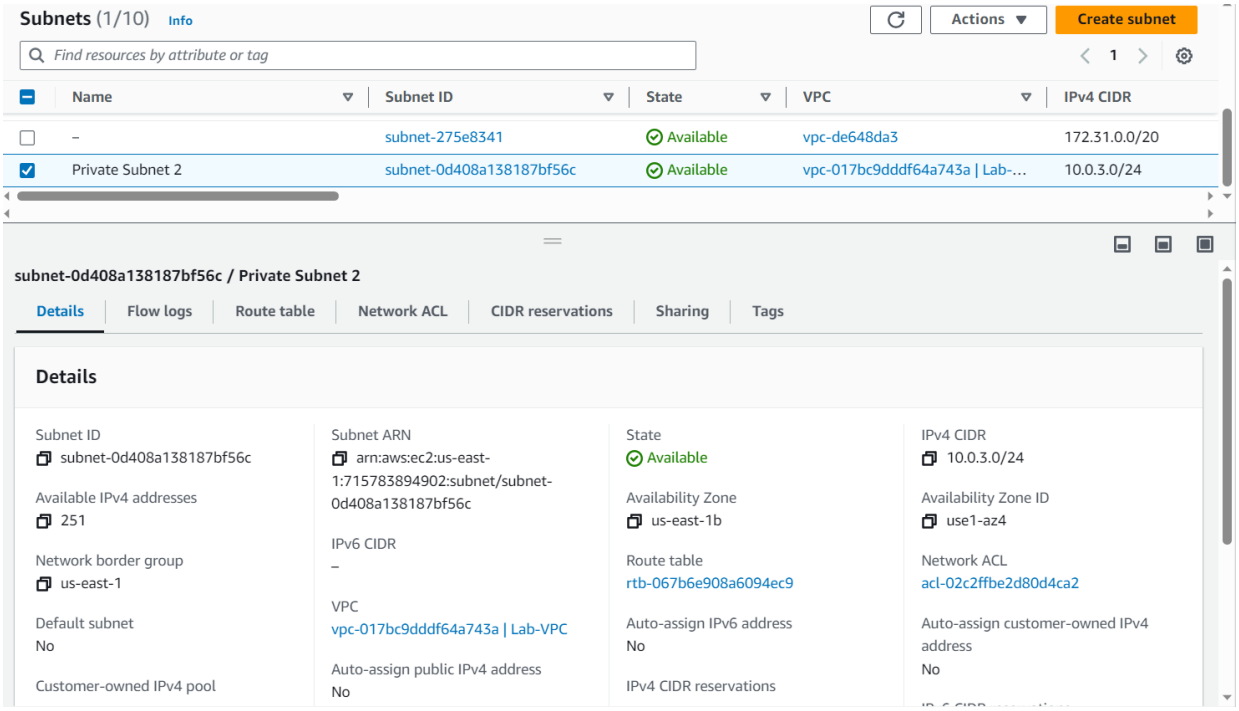
Auto-assign customer-owned IPv4 address
No

IPv6 CIDR reservations
–

Resource name DNS AAAA record
Disabled


Subnets (1/10) Info
Actions
Create subnet
Find resources by attribute or tag
1
Name
Subnet ID
State
VPC
IPv4 CIDR
Public Subnet 2
subnet-0d2285c7f14149ec8
Available
vpc-017bc9dddf64a743a | Lab-...
10.0.1.0/24
–
subnet-f97c10b4
Available
vpc-de648da3
172.31.16.0/20
subnet-0d2285c7f14149ec8 / Public Subnet 2
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Details
Subnet ID
subnet-0d2285c7f14149ec8
Subnet ARN
arn:aws:ec2:us-east-1:715783894902:subnet/subnet-0d2285c7f14149ec8
State
Available
IPv4 CIDR
10.0.1.0/24
Available IPv4 addresses
251
IPv6 CIDR
–
Availability Zone
us-east-1b
Availability Zone ID
use1-az4
Network border group
us-east-1
VPC
vpc-017bc9dddf64a743a | Lab-VPC
Route table
rtb-09f83c48ac045e565 | Lab Route Table
Network ACL
acl-02c2ffbe2d80d4ca2
Default subnet
No
Auto-assign public IPv4 address
No
Auto-assign IPv6 address
No
Auto-assign customer-owned IPv4 address
No
Customer-owned IPv4 pool


Subnets (1/10) Info
Actions
Create subnet
Find resources by attribute or tag
1
Name
Subnet ID
State
VPC
IPv4 CIDR
Private Subnet 1
subnet-0890c71dc174daccb
Available
vpc-017bc9dddf64a743a | Lab-...
10.0.2.0/24
–
subnet-86a37ad9
Available
vpc-de648da3
172.31.32.0/20
subnet-0890c71dc174daccb / Private Subnet 1
Details
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags
Details
Subnet ID
subnet-0890c71dc174daccb
Subnet ARN
arn:aws:ec2:us-east-1:715783894902:subnet/subnet-0890c71dc174daccb
State
Available
IPv4 CIDR
10.0.2.0/24
Available IPv4 addresses
251
IPv6 CIDR
–
Availability Zone
us-east-1a
Availability Zone ID
use1-az2
Network border group
us-east-1
VPC
vpc-017bc9dddf64a743a | Lab-VPC
Route table
rtb-067b6e908a6094ec9
Network ACL
acl-02c2ffbe2d80d4ca2
Default subnet
No
Auto-assign public IPv4 address
No
Auto-assign IPv6 address
No
Auto-assign customer-owned IPv4 address
No
Customer-owned IPv4 pool
IPv4 CIDR reservations

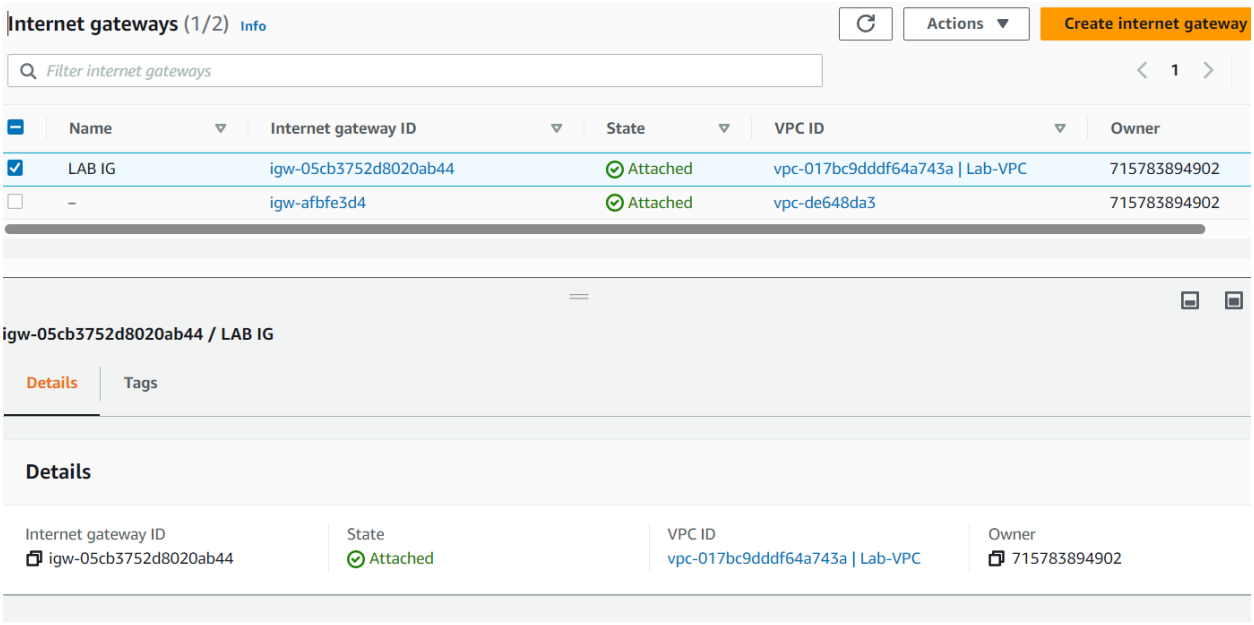


Internet Gateway

Code:

```
resource "aws_internet_gateway" "labgw" {
 vpc_id = aws_vpc.lab.id

 tags = {
   Name = "LAB IG"
 }
}
```

Result:

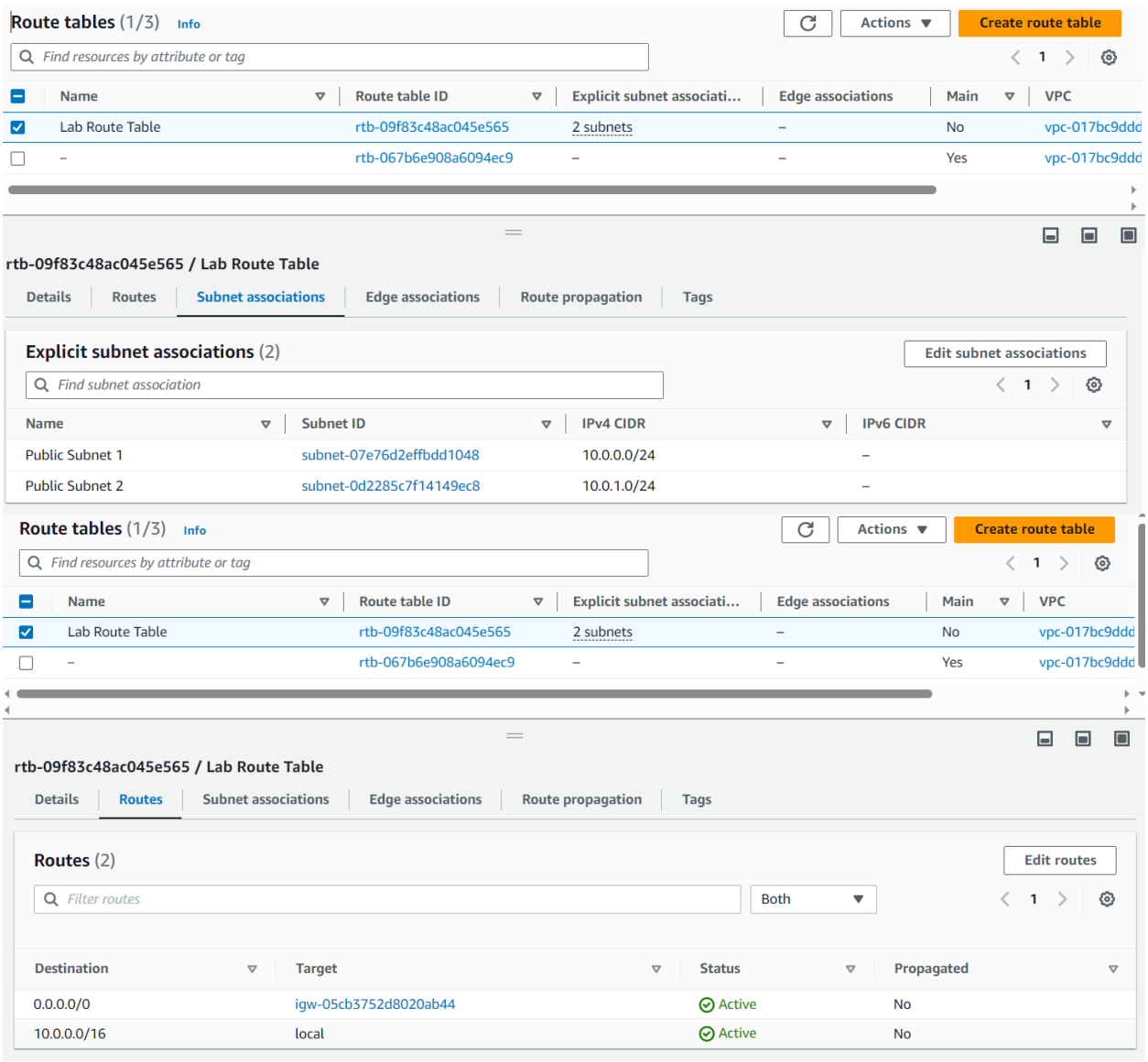


Route Table

Code:

```
resource "aws_route_table" "labrt" {
 vpc_id = aws_vpc.lab.id

 route {
  cidr_block = "0.0.0.0/0"
  gateway_id = aws_internet_gateway.labgw.id
 }

 tags = {
  Name = "Lab Route Table"
 }
}

resource "aws_route_table_association" "public_subnet1_assoc" {
 subnet_id      = aws_subnet.public_subnet1.id
 route_table_id = aws_route_table.labrt.id
}

resource "aws_route_table_association" "public_subnet2_assoc" {
 subnet_id      = aws_subnet.public_subnet2.id
 route_table_id = aws_route_table.labrt.id
}
```

Result:



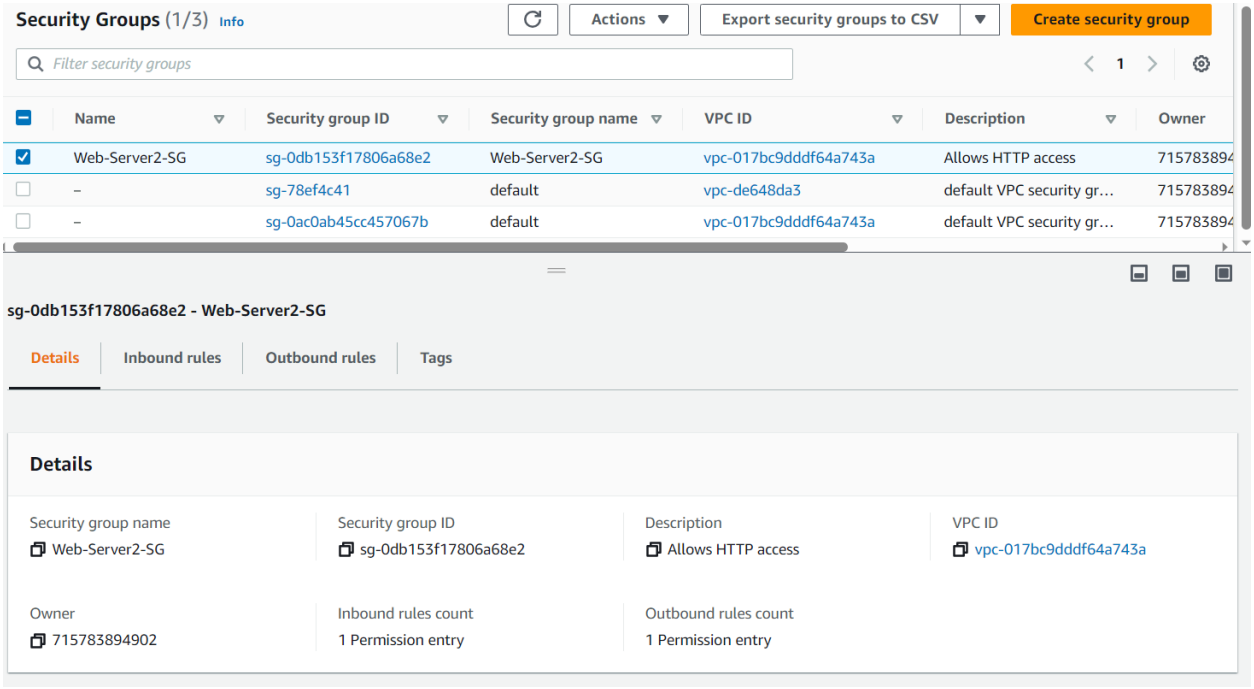
Security Group

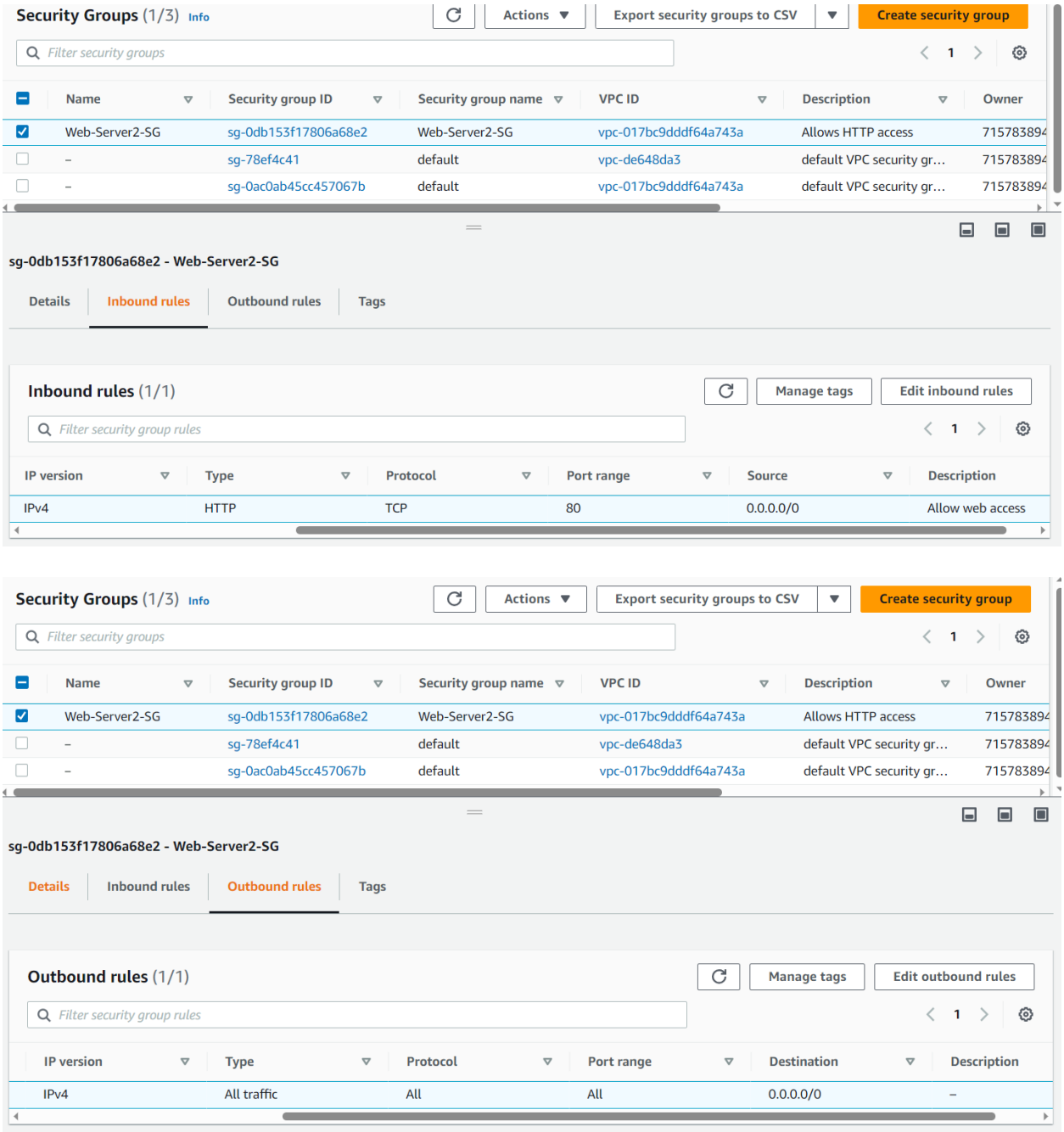
Code:

```
resource "aws_security_group" "webserversg2" {
  name        = "Web-Server2-SG"
  description = "Allows HTTP access"
  vpc_id      = aws_vpc.lab.id

  ingress {
    description = "Allow web access"
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

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

 tags = {
  Name = "Web-Server2-SG"
 }
}
```

Result:





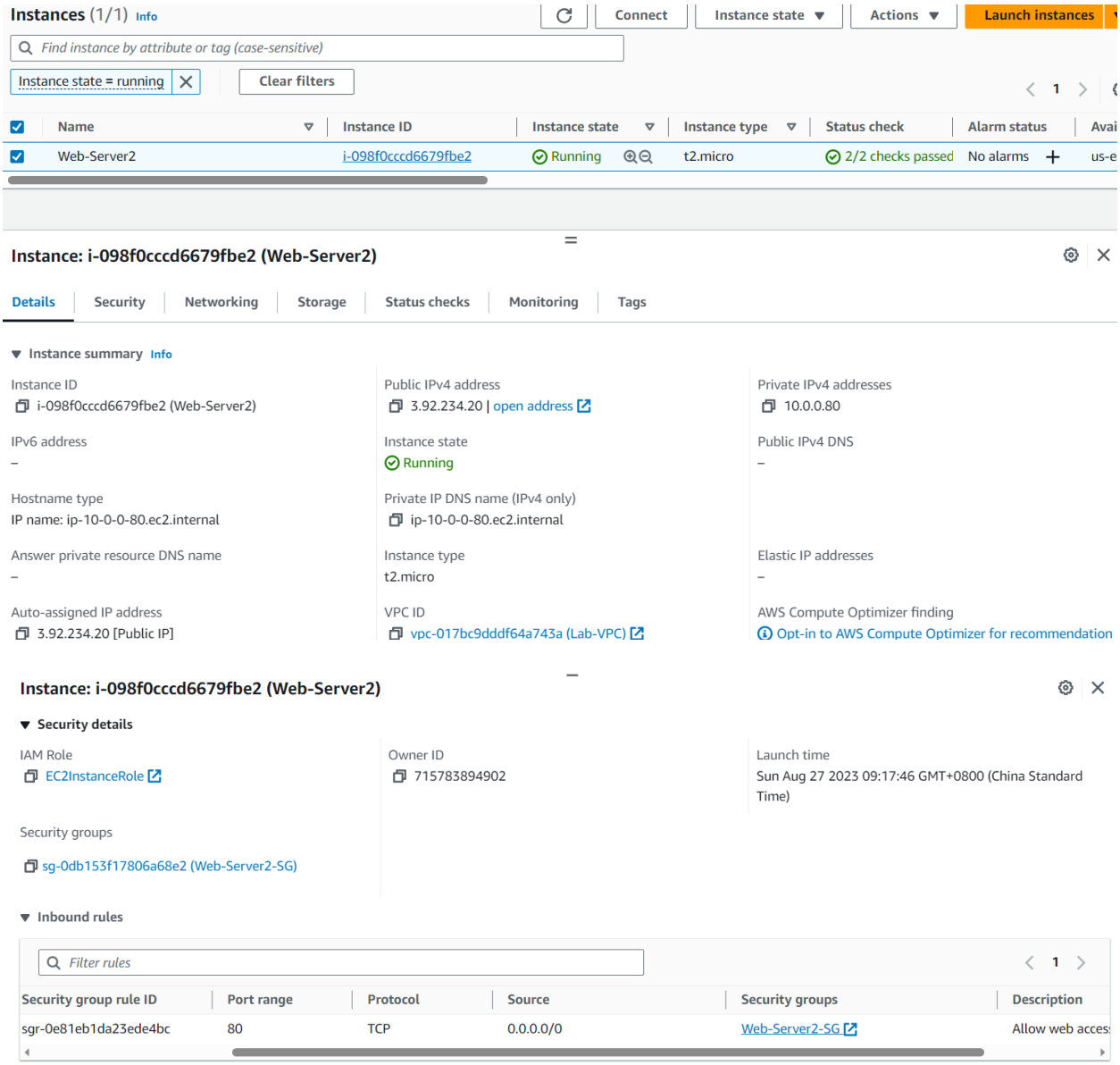
EC2 Instance

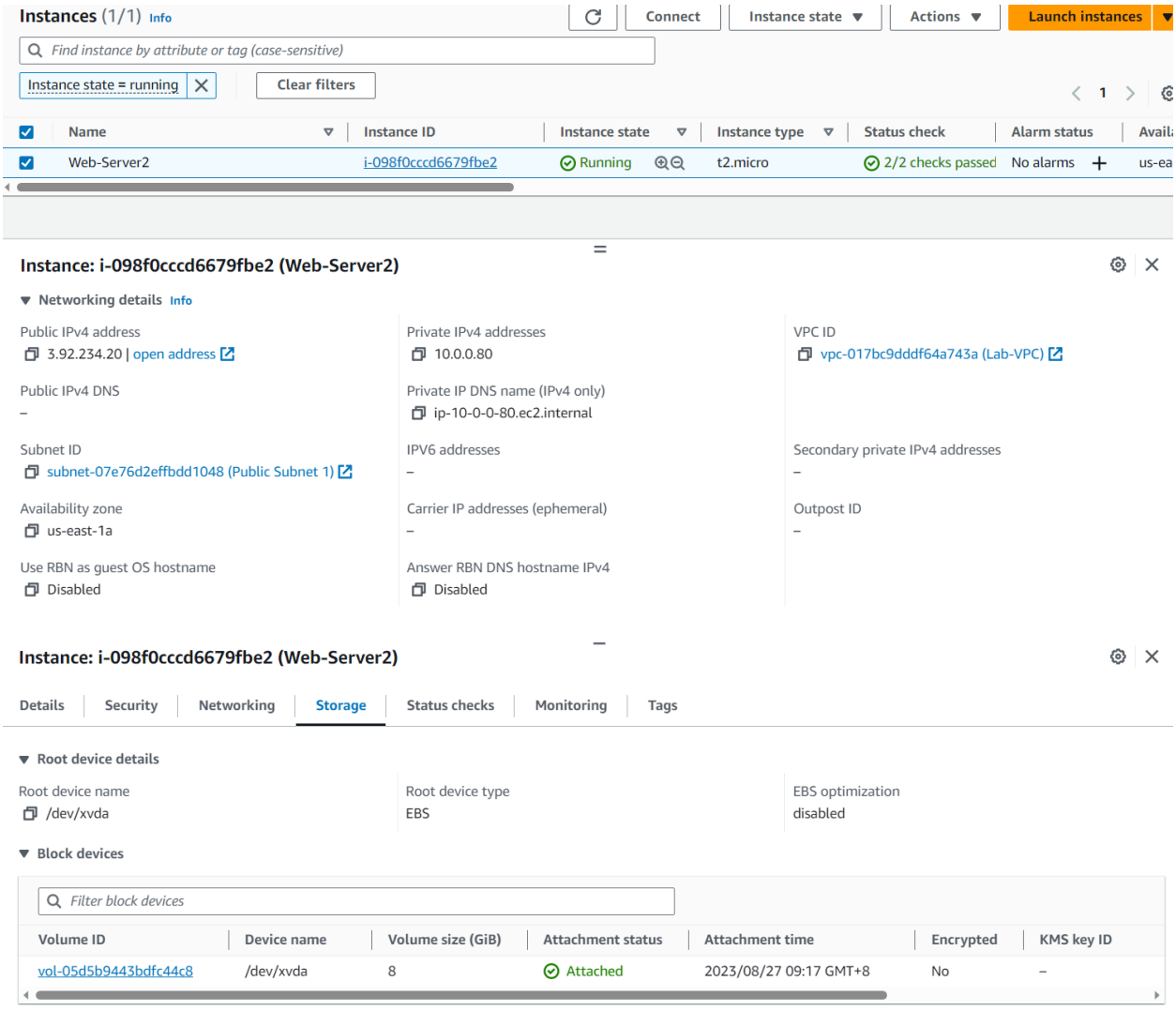
Code:

```
data "aws_ami" "amazon-linux-2" {
 most_recent = true
 owners = [ "amazon" ]
 filter {
   name   = "name"
   values = [ "amzn2-ami-hvm-*-gp2" ]
 }
}
```

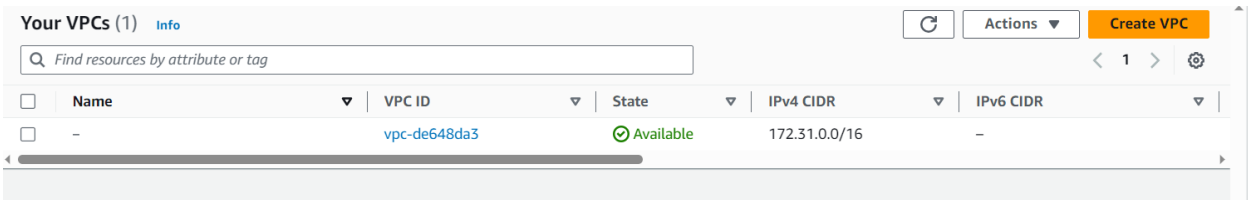
```
resource "aws_instance" "Web-Server2" {
  ami           = data.aws_ami.amazon-linux-2.id
  instance_type = "t2.micro"
  subnet_id = aws_subnet.public_subnet1.id
  associate_public_ip_address = true
  vpc_security_group_ids = [aws_security_group.webserversg2.id]
  tags = {
    Name = "Web-Server2"
  }
  iam_instance_profile = "Work-Role"
  user_data = <<EOF
  #!/bin/bash
  # Install Apache Web Server and PHP
  yum install -y httpd mysql
  amazon-linux-extras install -y php7.2
  # Download Lab files
  wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-EDNETW-1-60961/1-lab-
getting-started-vpc/s3/inventory-app.zip
  unzip inventory-app.zip -d /var/www/html/
  # Download and install the AWS SDK for PHP
  wget https://github.com/aws/aws-sdk-php/releases/download/3.62.3/aws.zip
  unzip aws -d /var/www/html
  # Turn on web server
  chkconfig httpd on
  service httpd start
  EOF
}
```

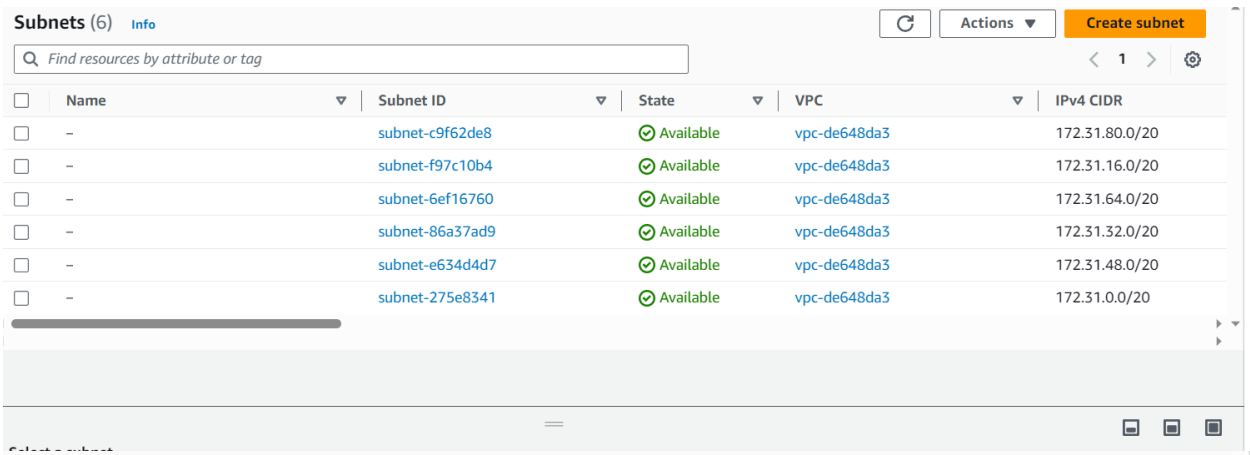
Result:

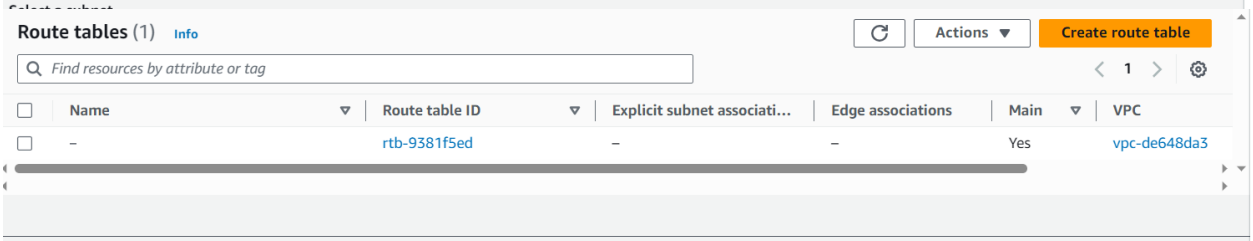

Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance state = running
Clear filters
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Web-Server2
i-098f0cccd6679fbe2
Running
t2.micro
2/2 checks passed
No alarms
Instance: i-098f0cccd6679fbe2 (Web-Server2)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-098f0cccd6679fbe2 (Web-Server2)
Public IPv4 address
3.92.234.20 | open address
Private IPv4 addresses
10.0.0.80
IPv6 address
–
Instance state
Running
Public IPv4 DNS
–
Hostname type
IP name: ip-10-0-0-80.ec2.internal
Private IP DNS name (IPv4 only)
ip-10-0-0-80.ec2.internal
Answer private resource DNS name
–
Instance type
t2.micro
Elastic IP addresses
–
Auto-assigned IP address
3.92.234.20 [Public IP]
VPC ID
vpc-017bc9dddf64a743a (Lab-VPC)
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendation
Instance: i-098f0cccd6679fbe2 (Web-Server2)
Security details
IAM Role
EC2InstanceRole
Owner ID
715783894902
Launch time
Sun Aug 27 2023 09:17:46 GMT+0800 (China Standard Time)
Security groups
sg-0db153f17806a68e2 (Web-Server2-SG)
Inbound rules
Filter rules
1
Security group rule ID
Port range
Protocol
Source
Security groups
Description
sgr-0e81eb1da23ede4bc
80
TCP
0.0.0.0/0
Web-Server2-SG
Allow web acces



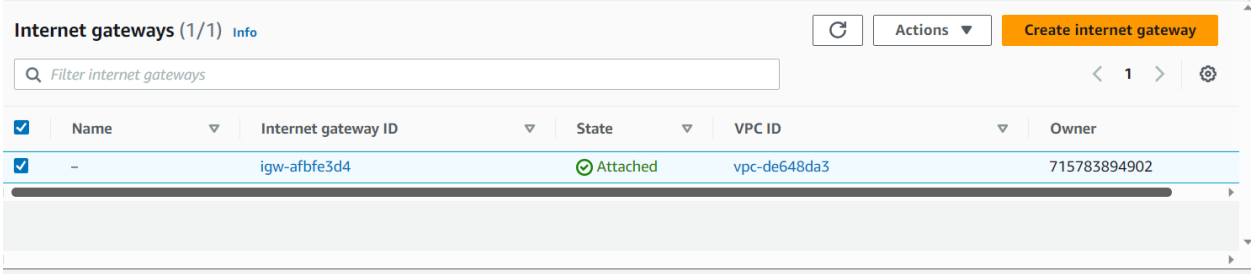
Web Server Result:







Select a route table

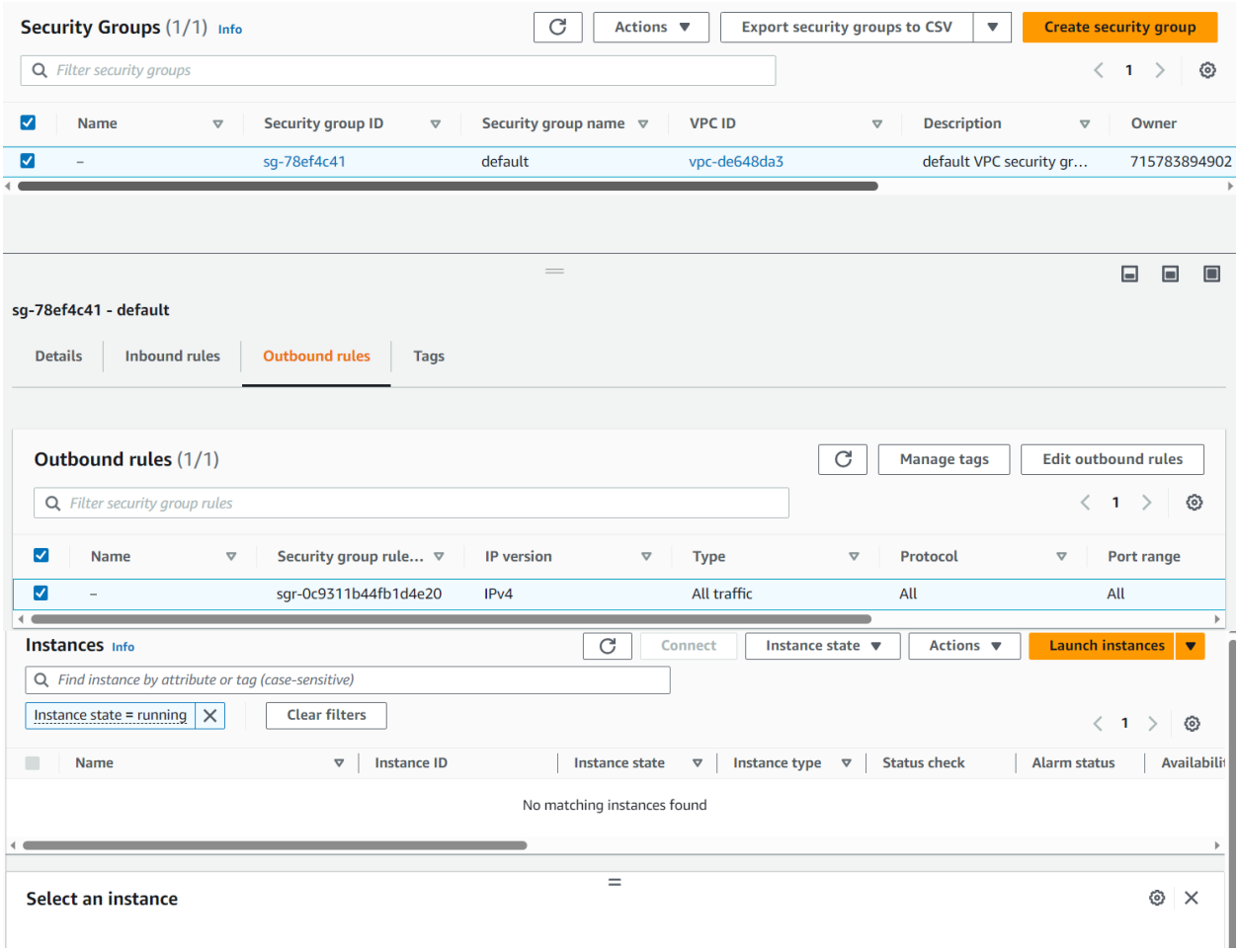


igw-afbfe3d4

Details | Tags

Details

Internet gateway ID	State	VPC ID	Owner
igw-afbfe3d4	Attached	vpc-de648da3	715783894902



Terraform Apply Log

```
D:\Terraform\terraform_aws\EC2>terraform apply
data.aws_ami.amazon-linux-2: Reading...
data.aws_ami.amazon-linux-2: Read complete after 2s [id=ami-0e1c5d8c23330dee3]

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.Web-Server2 will be created
  + resource "aws_instance" "Web-Server2" {
      + ami                          = "ami-0e1c5d8c23330dee3"
      + arn                          = (known after apply)
      + associate_public_ip_address  = true
      + 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)
  + get_password_data                  = false
  + host_id                            = (known after apply)
  + host_resource_group_arn            = (known after apply)
  + iam_instance_profile               = "Work-Role"
  + id                                 = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle                 = (known after apply)
  + instance_state                     = (known after apply)
  + instance_type                      = "t2.micro"
  + ipv6_address_count                 = (known after apply)
  + ipv6_addresses                     = (known after apply)
  + key_name                           = (known after apply)
  + monitoring                         = (known after apply)
  + outpost_arn                        = (known after apply)
  + password_data                      = (known after apply)
  + placement_group                    = (known after apply)
  + placement_partition_number         = (known after apply)
  + primary_network_interface_id       = (known after apply)
  + private_dns                        = (known after apply)
  + private_ip                         = (known after apply)
  + public_dns                         = (known after apply)
  + public_ip                          = (known after apply)
  + secondary_private_ips              = (known after apply)
  + security_groups                    = (known after apply)
  + source_dest_check                  = true
  + spot_instance_request_id           = (known after apply)
  + subnet_id                          = (known after apply)
  + tags                               = {
      + "Name" = "Web-Server2"
    }
  + tags_all                           = {
      + "Name" = "Web-Server2"
    }
  + tenancy                            = (known after apply)
  + user_data                          = "abdb6c5b0745d074e2d2210932d7f41e56c27052"
  + user_data_base64                   = (known after apply)
  + user_data_replace_on_change        = false
  + vpc_security_group_ids             = (known after apply)
 }

# aws_internet_gateway.labgw will be created
+ resource "aws_internet_gateway" "labgw" {
  + arn      = (known after apply)
  + id       = (known after apply)
  + owner_id = (known after apply)
```

```
    + tags     = {
        + "Name" = "LAB IG"
      }
    + tags_all = {
        + "Name" = "LAB IG"
      }
    + vpc_id   = (known after apply)
  }

# aws_route_table.labrt will be created
+ resource "aws_route_table" "labrt" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + carrier_gateway_id        = ""
            + cidr_block                = "0.0.0.0/0"
            + core_network_arn          = ""
            + destination_prefix_list_id = ""
            + egress_only_gateway_id    = ""
            + gateway_id                = (known after apply)
            + ipv6_cidr_block           = ""
            + local_gateway_id          = ""
            + nat_gateway_id            = ""
            + network_interface_id      = ""
            + transit_gateway_id        = ""
            + vpc_endpoint_id           = ""
            + vpc_peering_connection_id = ""
          },
      ]
    + tags             = {
        + "Name" = "Lab Route Table"
      }
    + tags_all         = {
        + "Name" = "Lab Route Table"
      }
    + vpc_id           = (known after apply)
  }

# aws_route_table_association.public_subnet1_assoc will be created
+ resource "aws_route_table_association" "public_subnet1_assoc" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }
```

```
# aws_route_table_association.public_subnet2_assoc will be created
+ resource "aws_route_table_association" "public_subnet2_assoc" {
   + id           = (known after apply)
   + route_table_id = (known after apply)
   + subnet_id    = (known after apply)
  }

# aws_security_group.webserversg2 will be created
+ resource "aws_security_group" "webserversg2" {
   + arn                  = (known after apply)
   + description          = "Allows HTTP access"
   + egress               = [
      + {
         + cidr_blocks     = [
            + "0.0.0.0/0",
          ]
         + description     = ""
         + from_port       = 0
         + ipv6_cidr_blocks = []
         + prefix_list_ids  = []
         + protocol        = "-1"
         + security_groups  = []
         + self            = false
         + to_port         = 0
        },
    ]
   + id                   = (known after apply)
   + ingress              = [
      + {
         + cidr_blocks     = [
            + "0.0.0.0/0",
          ]
         + description     = "Allow web access"
         + from_port       = 80
         + ipv6_cidr_blocks = []
         + prefix_list_ids  = []
         + protocol        = "tcp"
         + security_groups  = []
         + self            = false
         + to_port         = 80
        },
    ]
   + name                 = "Web-Server2-SG"
   + name_prefix          = (known after apply)
   + owner_id             = (known after apply)
   + revoke_rules_on_delete = false
   + tags                 = {
      + "Name" = "Web-Server2-SG"
```

```
    }
  + tags_all             = {
      + "Name" = "Web-Server2-SG"
    }
  + vpc_id               = (known after apply)
 }

# aws_subnet.private_subnet1 will be created
+ resource "aws_subnet" "private_subnet1" {
  + arn                                            = (known after apply)
  + assign_ipv6_address_on_creation                = false
  + availability_zone                              = "us-east-1a"
  + availability_zone_id                           = (known after apply)
  + cidr_block                                     = "10.0.2.0/24"
  + enable_dns64                                   = false
  + enable_resource_name_dns_a_record_on_launch    = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id                                             = (known after apply)
  + ipv6_cidr_block_association_id                 = (known after apply)
  + ipv6_native                                    = false
  + map_public_ip_on_launch                        = false
  + owner_id                                       = (known after apply)
  + private_dns_hostname_type_on_launch            = (known after apply)
  + tags                                           = {
      + "Name" = "Private Subnet 1"
    }
  + tags_all                                       = {
      + "Name" = "Private Subnet 1"
    }
  + vpc_id                                         = (known after apply)
 }

# aws_subnet.private_subnet2 will be created
+ resource "aws_subnet" "private_subnet2" {
  + arn                                            = (known after apply)
  + assign_ipv6_address_on_creation                = false
  + availability_zone                              = "us-east-1b"
  + availability_zone_id                           = (known after apply)
  + cidr_block                                     = "10.0.3.0/24"
  + enable_dns64                                   = false
  + enable_resource_name_dns_a_record_on_launch    = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id                                             = (known after apply)
  + ipv6_cidr_block_association_id                 = (known after apply)
  + ipv6_native                                    = false
  + map_public_ip_on_launch                        = false
  + owner_id                                       = (known after apply)
  + private_dns_hostname_type_on_launch            = (known after apply)
```

```
    + tags                                  = {
        + "Name" = "Private Subnet 2"
      }
    + tags_all                              = {
        + "Name" = "Private Subnet 2"
      }
    + vpc_id                                = (known after apply)
  }

# aws_subnet.public_subnet1 will be created
+ resource "aws_subnet" "public_subnet1" {
    + arn                                   = (known after apply)
    + assign_ipv6_address_on_creation       = false
    + availability_zone                     = "us-east-1a"
    + availability_zone_id                  = (known after apply)
    + cidr_block                            = "10.0.0.0/24"
    + enable_dns64                          = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                    = (known after apply)
    + ipv6_cidr_block_association_id        = (known after apply)
    + ipv6_native                           = false
    + map_public_ip_on_launch               = false
    + owner_id                              = (known after apply)
    + private_dns_hostname_type_on_launch   = (known after apply)
    + tags                                  = {
        + "Name" = "Public Subnet 1"
      }
    + tags_all                              = {
        + "Name" = "Public Subnet 1"
      }
    + vpc_id                                = (known after apply)
  }

# aws_subnet.public_subnet2 will be created
+ resource "aws_subnet" "public_subnet2" {
    + arn                                   = (known after apply)
    + assign_ipv6_address_on_creation       = false
    + availability_zone                     = "us-east-1b"
    + availability_zone_id                  = (known after apply)
    + cidr_block                            = "10.0.1.0/24"
    + enable_dns64                          = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                    = (known after apply)
    + ipv6_cidr_block_association_id        = (known after apply)
    + ipv6_native                           = false
    + map_public_ip_on_launch               = false
```

```
    + owner_id                                     = (known after apply)
    + private_dns_hostname_type_on_launch          = (known after apply)
    + tags                                         = {
        + "Name" = "Public Subnet 2"
      }
    + tags_all                                     = {
        + "Name" = "Public Subnet 2"
      }
    + vpc_id                                       = (known after apply)
  }

  # aws_vpc.lab will be created
  + resource "aws_vpc" "lab" {
    + arn                                = (known after apply)
    + cidr_block                         = "10.0.0.0/16"
    + default_network_acl_id             = (known after apply)
    + default_route_table_id             = (known after apply)
    + default_security_group_id          = (known after apply)
    + dhcp_options_id                    = (known after apply)
    + enable_dns_hostnames               = (known after apply)
    + enable_dns_support                 = true
    + enable_network_address_usage_metrics = (known after apply)
    + id                                 = (known after apply)
    + instance_tenancy                   = "default"
    + ipv6_association_id                = (known after apply)
    + ipv6_cidr_block                    = (known after apply)
    + ipv6_cidr_block_network_border_group = (known after apply)
    + main_route_table_id                = (known after apply)
    + owner_id                           = (known after apply)
    + tags                               = {
        + "Name" = "Lab-VPC"
      }
    + tags_all                           = {
        + "Name" = "Lab-VPC"
      }
  }

Plan: 11 to add, 0 to change, 0 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_vpc.lab: Creating...
aws_vpc.lab: Creation complete after 3s [id=vpc-017bc9dddf64a743a]
aws_internet_gateway.labgw: Creating...
```

aws_subnet.public_subnet2: Creating...
aws_subnet.public_subnet1: Creating...
aws_subnet.private_subnet2: Creating...
aws_subnet.private_subnet1: Creating...
aws_security_group.webserversg2: Creating...
aws_subnet.public_subnet1: Creation complete after 2s [id=subnet-07e76d2effbdd1048]
aws_subnet.private_subnet1: Creation complete after 2s [id=subnet-0890c71dc174daccb]
aws_subnet.private_subnet2: Creation complete after 2s [id=subnet-0d408a138187bf56c]
aws_internet_gateway.labgw: Creation complete after 3s [id=igw-05cb3752d8020ab44]
aws_subnet.public_subnet2: Creation complete after 3s [id=subnet-0d2285c7f14149ec8]
aws_route_table.labrt: Creating...
aws_route_table.labrt: Creation complete after 2s [id=rtb-09f83c48ac045e565]
aws_route_table_association.public_subnet2_assoc: Creating...
aws_route_table_association.public_subnet1_assoc: Creating...
aws_route_table_association.public_subnet1_assoc: Creation complete after 1s [id=rtbassoc-0899e4118a34fbc18]
aws_route_table_association.public_subnet2_assoc: Creation complete after 1s [id=rtbassoc-0b30a42bbedefdeeb]
aws_security_group.webserversg2: Creation complete after 7s [id=sg-0db153f17806a68e2]
aws_instance.Web-Server2: Creating...
aws_instance.Web-Server2: Still creating... [10s elapsed]
aws_instance.Web-Server2: Still creating... [20s elapsed]
aws_instance.Web-Server2: Still creating... [30s elapsed]
aws_instance.Web-Server2: Creation complete after 35s [id=i-098f0cccd6679fbe2]

Apply complete! Resources: 11 added, 0 changed, 0 destroyed.

D:\Terraform\terraform_aws\EC2>

Terraform Destroy Log

Destroy Result:
D:\Terraform\terraform_aws\EC2>terraform destroy
aws_route_table_association.public_subnet2_assoc: Refreshing state... [id=rtbassoc-0b30a42bbedefdeeb]
aws_vpc.lab: Refreshing state... [id=vpc-017bc9dddf64a743a]
data.aws_ami.amazon-linux-2: Reading...
data.aws_ami.amazon-linux-2: Read complete after 2s [id=ami-0e1c5d8c23330dee3]
aws_subnet.private_subnet2: Refreshing state... [id=subnet-0d408a138187bf56c]
aws_internet_gateway.labgw: Refreshing state... [id=igw-05cb3752d8020ab44]
aws_subnet.public_subnet1: Refreshing state... [id=subnet-07e76d2effbdd1048]
aws_subnet.public_subnet2: Refreshing state... [id=subnet-0d2285c7f14149ec8]
aws_subnet.private_subnet1: Refreshing state... [id=subnet-0890c71dc174daccb]
aws_security_group.webserversg2: Refreshing state... [id=sg-0db153f17806a68e2]
aws_route_table.labrt: Refreshing state... [id=rtb-09f83c48ac045e565]
aws_instance.Web-Server2: Refreshing state... [id=i-098f0cccd6679fbe2]

aws_route_table_association.public_subnet1_assoc: Refreshing state... [id=rtbassoc-0899e4118a34fbc18]

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_instance.Web-Server2 will be destroyed
  - resource "aws_instance" "Web-Server2" {
      - ami                          = "ami-0e1c5d8c23330dee3" -> null
      - arn                          = "arn:aws:ec2:us-east-1:715783894902:instance/i-098f0cccd6679fbe2" -> null
      - associate_public_ip_address  = true -> null
      - availability_zone            = "us-east-1a" -> null
      - cpu_core_count               = 1 -> null
      - cpu_threads_per_core         = 1 -> null
      - disable_api_stop             = false -> null
      - disable_api_termination      = false -> null
      - ebs_optimized                = false -> null
      - get_password_data            = false -> null
      - hibernation                  = false -> null
      - iam_instance_profile         = "Work-Role" -> null
      - id                           = "i-098f0cccd6679fbe2" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state               = "running" -> null
      - instance_type                = "t2.micro" -> null
      - ipv6_address_count           = 0 -> null
      - ipv6_addresses               = [] -> null
      - monitoring                   = false -> null
      - placement_partition_number   = 0 -> null
      - primary_network_interface_id = "eni-0104387af0cab9f05" -> null
      - private_dns                  = "ip-10-0-0-80.ec2.internal" -> null
      - private_ip                   = "10.0.0.80" -> null
      - public_ip                    = "3.92.234.20" -> null
      - secondary_private_ips        = [] -> null
      - security_groups              = [] -> null
      - source_dest_check            = true -> null
      - subnet_id                    = "subnet-07e76d2effbdd1048" -> null
      - tags                         = {
          - "Name" = "Web-Server2"
        } -> null
      - tags_all                     = {
          - "Name" = "Web-Server2"
        } -> null
      - tenancy                      = "default" -> null
      - user_data                    = "abdb6c5b0745d074e2d2210932d7f41e56c27052" -> null
```

```
- user_data_replace_on_change         = false -> null
- vpc_security_group_ids              = [
   - "sg-0db153f17806a68e2",
  ] -> null

- capacity_reservation_specification {
   - capacity_reservation_preference = "open" -> null
  }

- cpu_options {
   - core_count       = 1 -> null
   - threads_per_core = 1 -> null
  }

- credit_specification {
   - cpu_credits = "standard" -> null
  }

- enclave_options {
   - enabled = false -> null
  }

- maintenance_options {
   - auto_recovery = "default" -> null
  }

- metadata_options {
   - http_endpoint              = "enabled" -> null
   - http_protocol_ipv6         = "disabled" -> null
   - http_put_response_hop_limit = 1 -> null
   - http_tokens                = "optional" -> null
   - instance_metadata_tags     = "disabled" -> null
  }

- private_dns_name_options {
   - enable_resource_name_dns_a_record    = false -> null
   - enable_resource_name_dns_aaaa_record = false -> null
   - hostname_type                        = "ip-name" -> null
  }

- root_block_device {
   - delete_on_termination = true -> null
   - device_name           = "/dev/xvda" -> null
   - encrypted             = false -> null
   - iops                  = 100 -> null
   - tags                  = {} -> null
   - throughput            = 0 -> null
   - volume_id             = "vol-05d5b9443bdfc44c8" -> null
```

```
      - volume_size          = 8 -> null
      - volume_type          = "gp2" -> null
    }
  }

# aws_internet_gateway.labgw will be destroyed
- resource "aws_internet_gateway" "labgw" {
   - arn      = "arn:aws:ec2:us-east-1:715783894902:internet-gateway/igw-05cb3752d8020ab44" -> null
   - id       = "igw-05cb3752d8020ab44" -> null
   - owner_id = "715783894902" -> null
   - tags     = {
      - "Name" = "LAB IG"
     } -> null
   - tags_all = {
      - "Name" = "LAB IG"
     } -> null
   - vpc_id   = "vpc-017bc9dddf64a743a" -> null
  }

# aws_route_table.labrt will be destroyed
- resource "aws_route_table" "labrt" {
   - arn              = "arn:aws:ec2:us-east-1:715783894902:route-table/rtb-09f83c48ac045e565" -> null
   - id               = "rtb-09f83c48ac045e565" -> null
   - owner_id         = "715783894902" -> null
   - propagating_vgws = [] -> null
   - route            = [
      - {
         - carrier_gateway_id        = ""
         - cidr_block                = "0.0.0.0/0"
         - core_network_arn          = ""
         - destination_prefix_list_id = ""
         - egress_only_gateway_id    = ""
         - gateway_id                = "igw-05cb3752d8020ab44"
         - ipv6_cidr_block           = ""
         - local_gateway_id          = ""
         - nat_gateway_id            = ""
         - network_interface_id      = ""
         - transit_gateway_id        = ""
         - vpc_endpoint_id           = ""
         - vpc_peering_connection_id  = ""
        },
     ] -> null
   - tags             = {
      - "Name" = "Lab Route Table"
     } -> null
   - tags_all         = {
      - "Name" = "Lab Route Table"
     } -> null
```

```
    - vpc_id          = "vpc-017bc9dddf64a743a" -> null
  }

  # aws_route_table_association.public_subnet1_assoc will be destroyed
  - resource "aws_route_table_association" "public_subnet1_assoc" {
    - id           = "rtbassoc-0899e4118a34fbc18" -> null
    - route_table_id = "rtb-09f83c48ac045e565" -> null
    - subnet_id      = "subnet-07e76d2effbdd1048" -> null
  }

  # aws_route_table_association.public_subnet2_assoc will be destroyed
  - resource "aws_route_table_association" "public_subnet2_assoc" {
    - id           = "rtbassoc-0b30a42bbedefdeeb" -> null
    - route_table_id = "rtb-09f83c48ac045e565" -> null
    - subnet_id      = "subnet-0d2285c7f14149ec8" -> null
  }

  # aws_security_group.webserversg2 will be destroyed
  - resource "aws_security_group" "webserversg2" {
    - arn                  = "arn:aws:ec2:us-east-1:715783894902:security-group/sg-0db153f17806a68e2" ->
null
    - description          = "Allows HTTP access" -> null
    - egress               = [
       - {
          - cidr_blocks     = [
             - "0.0.0.0/0",
            ]
          - description     = ""
          - from_port       = 0
          - ipv6_cidr_blocks = []
          - prefix_list_ids  = []
          - protocol        = "-1"
          - security_groups  = []
          - self            = false
          - to_port         = 0
         },
      ] -> null
    - id                   = "sg-0db153f17806a68e2" -> null
    - ingress              = [
       - {
          - cidr_blocks     = [
             - "0.0.0.0/0",
            ]
          - description     = "Allow web access"
          - from_port       = 80
          - ipv6_cidr_blocks = []
          - prefix_list_ids  = []
          - protocol        = "tcp"
```

```
          - security_groups  = []
          - self           = false
          - to_port        = 80
         },
      ] -> null
    - name                = "Web-Server2-SG" -> null
    - owner_id            = "715783894902" -> null
    - revoke_rules_on_delete = false -> null
    - tags                = {
       - "Name" = "Web-Server2-SG"
      } -> null
    - tags_all            = {
       - "Name" = "Web-Server2-SG"
      } -> null
    - vpc_id              = "vpc-017bc9dddf64a743a" -> null
  }

  # aws_subnet.private_subnet1 will be destroyed
  - resource "aws_subnet" "private_subnet1" {
    - arn                                = "arn:aws:ec2:us-east-1:715783894902:subnet/subnet-
0890c71dc174daccb" -> null
    - assign_ipv6_address_on_creation              = false -> null
    - availability_zone                  = "us-east-1a" -> null
    - availability_zone_id                 = "use1-az2" -> null
    - cidr_block                        = "10.0.2.0/24" -> null
    - enable_dns64                       = false -> null
    - enable_lni_at_device_index               = 0 -> null
    - enable_resource_name_dns_a_record_on_launch    = false -> null
    - enable_resource_name_dns_aaaa_record_on_launch = false -> null
    - id                                 = "subnet-0890c71dc174daccb" -> null
    - ipv6_native                        = false -> null
    - map_customer_owned_ip_on_launch            = false -> null
    - map_public_ip_on_launch                  = false -> null
    - owner_id                           = "715783894902" -> null
    - private_dns_hostname_type_on_launch          = "ip-name" -> null
    - tags                               = {
       - "Name" = "Private Subnet 1"
      } -> null
    - tags_all                           = {
       - "Name" = "Private Subnet 1"
      } -> null
    - vpc_id                             = "vpc-017bc9dddf64a743a" -> null
  }

  # aws_subnet.private_subnet2 will be destroyed
  - resource "aws_subnet" "private_subnet2" {
    - arn                                = "arn:aws:ec2:us-east-1:715783894902:subnet/subnet-
0d408a138187bf56c" -> null
```

```
    - assign_ipv6_address_on_creation              = false -> null
    - availability_zone                            = "us-east-1b" -> null
    - availability_zone_id                         = "use1-az4" -> null
    - cidr_block                                   = "10.0.3.0/24" -> null
    - enable_dns64                                 = false -> null
    - enable_lni_at_device_index                   = 0 -> null
    - enable_resource_name_dns_a_record_on_launch  = false -> null
    - enable_resource_name_dns_aaaa_record_on_launch = false -> null
    - id                                           = "subnet-0d408a138187bf56c" -> null
    - ipv6_native                                  = false -> null
    - map_customer_owned_ip_on_launch              = false -> null
    - map_public_ip_on_launch                      = false -> null
    - owner_id                                     = "715783894902" -> null
    - private_dns_hostname_type_on_launch          = "ip-name" -> null
    - tags                                         = {
        - "Name" = "Private Subnet 2"
      } -> null
    - tags_all                                     = {
        - "Name" = "Private Subnet 2"
      } -> null
    - vpc_id                                       = "vpc-017bc9dddf64a743a" -> null
  }

  # aws_subnet.public_subnet1 will be destroyed
  - resource "aws_subnet" "public_subnet1" {
    - arn                                          = "arn:aws:ec2:us-east-1:715783894902:subnet/subnet-
07e76d2effbdd1048" -> null
    - assign_ipv6_address_on_creation              = false -> null
    - availability_zone                            = "us-east-1a" -> null
    - availability_zone_id                         = "use1-az2" -> null
    - cidr_block                                   = "10.0.0.0/24" -> null
    - enable_dns64                                 = false -> null
    - enable_lni_at_device_index                   = 0 -> null
    - enable_resource_name_dns_a_record_on_launch  = false -> null
    - enable_resource_name_dns_aaaa_record_on_launch = false -> null
    - id                                           = "subnet-07e76d2effbdd1048" -> null
    - ipv6_native                                  = false -> null
    - map_customer_owned_ip_on_launch              = false -> null
    - map_public_ip_on_launch                      = false -> null
    - owner_id                                     = "715783894902" -> null
    - private_dns_hostname_type_on_launch          = "ip-name" -> null
    - tags                                         = {
        - "Name" = "Public Subnet 1"
      } -> null
    - tags_all                                     = {
        - "Name" = "Public Subnet 1"
      } -> null
    - vpc_id                                       = "vpc-017bc9dddf64a743a" -> null
```

```
  }

  # aws_subnet.public_subnet2 will be destroyed
  - resource "aws_subnet" "public_subnet2" {
      - arn                                            = "arn:aws:ec2:us-east-1:715783894902:subnet/subnet-
0d2285c7f14149ec8" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "us-east-1b" -> null
      - availability_zone_id                           = "use1-az4" -> null
      - cidr_block                                     = "10.0.1.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-0d2285c7f14149ec8" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "715783894902" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - tags                                           = {
          - "Name" = "Public Subnet 2"
        } -> null
      - tags_all                                       = {
          - "Name" = "Public Subnet 2"
        } -> null
      - vpc_id                                         = "vpc-017bc9dddf64a743a" -> null
    }

  # aws_vpc.lab will be destroyed
  - resource "aws_vpc" "lab" {
      - arn                                  = "arn:aws:ec2:us-east-1:715783894902:vpc/vpc-017bc9dddf64a743a" -> null
      - assign_generated_ipv6_cidr_block     = false -> null
      - cidr_block                           = "10.0.0.0/16" -> null
      - default_network_acl_id               = "acl-02c2ffbe2d80d4ca2" -> null
      - default_route_table_id               = "rtb-067b6e908a6094ec9" -> null
      - default_security_group_id            = "sg-0ac0ab45cc457067b" -> null
      - dhcp_options_id                      = "dopt-5861eb22" -> null
      - enable_dns_hostnames                 = false -> null
      - enable_dns_support                   = true -> null
      - enable_network_address_usage_metrics = false -> null
      - id                                   = "vpc-017bc9dddf64a743a" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
      - main_route_table_id                  = "rtb-067b6e908a6094ec9" -> null
      - owner_id                             = "715783894902" -> null
      - tags                                 = {
          - "Name" = "Lab-VPC"
```

```
    } -> null
  - tags_all                        = {
      - "Name" = "Lab-VPC"
    } -> null
}

Plan: 0 to add, 0 to change, 11 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_route_table_association.public_subnet2_assoc: Destroying... [id=rtbassoc-0b30a42bbedefdeeb]
aws_route_table_association.public_subnet1_assoc: Destroying... [id=rtbassoc-0899e4118a34fbc18]
aws_subnet.private_subnet2: Destroying... [id=subnet-0d408a138187bf56c]
aws_subnet.private_subnet1: Destroying... [id=subnet-0890c71dc174daccb]
aws_instance.Web-Server2: Destroying... [id=i-098f0cccd6679fbe2]
aws_route_table_association.public_subnet2_assoc: Destruction complete after 1s
aws_subnet.public_subnet2: Destroying... [id=subnet-0d2285c7f14149ec8]
aws_route_table_association.public_subnet1_assoc: Destruction complete after 1s
aws_route_table.labrt: Destroying... [id=rtb-09f83c48ac045e565]
aws_subnet.private_subnet2: Destruction complete after 2s
aws_subnet.private_subnet1: Destruction complete after 2s
aws_subnet.public_subnet2: Destruction complete after 1s
aws_route_table.labrt: Destruction complete after 2s
aws_internet_gateway.labgw: Destroying... [id=igw-05cb3752d8020ab44]
aws_instance.Web-Server2: Still destroying... [id=i-098f0cccd6679fbe2, 10s elapsed]
aws_internet_gateway.labgw: Still destroying... [id=igw-05cb3752d8020ab44, 10s elapsed]
aws_instance.Web-Server2: Still destroying... [id=i-098f0cccd6679fbe2, 20s elapsed]
aws_internet_gateway.labgw: Still destroying... [id=igw-05cb3752d8020ab44, 20s elapsed]
aws_instance.Web-Server2: Still destroying... [id=i-098f0cccd6679fbe2, 30s elapsed]
aws_internet_gateway.labgw: Still destroying... [id=igw-05cb3752d8020ab44, 30s elapsed]
aws_instance.Web-Server2: Still destroying... [id=i-098f0cccd6679fbe2, 40s elapsed]
aws_internet_gateway.labgw: Destruction complete after 39s
aws_instance.Web-Server2: Destruction complete after 43s
aws_subnet.public_subnet1: Destroying... [id=subnet-07e76d2effbdd1048]
aws_security_group.webserversg2: Destroying... [id=sg-0db153f17806a68e2]
aws_subnet.public_subnet1: Destruction complete after 1s
aws_security_group.webserversg2: Destruction complete after 1s
aws_vpc.lab: Destroying... [id=vpc-017bc9dddf64a743a]
aws_vpc.lab: Destruction complete after 1s

Destroy complete! Resources: 11 destroyed.

D:\Terraform\terraform_aws\EC2>
```